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

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 August 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 natural 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 to 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. $15''$) but represents land use/land cover (LULC) components as sub-pixel fractions rather than discrete categories. After aggregating to the $5'$ grid we present a method for decomposing the natural vegetation/cropland mosaic class found in MLCT into constituent classes as a function of the local landscape. We then 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 resulting data model is conceptually equivalent to a stack of spectral bands with the additional property that the components of each pixel sum to unity. Its classification scheme 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.

Acknowledgments

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/`) project.

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**

Chapter 1 Introduction **1**
1.1 Background 1
1.2 Objective 3
1.3 Reproducible Research 6

Chapter 2 Data Sets and Methodology **9**
2.1 MODIS Land Cover Type (MLCT) 10
2.1.1 Reclassification 10
2.1.2 Aggregation 14
2.1.3 Mosaic decomposition 30
2.2 National Land-cover Database 2001 (NLCD) 34
2.2.1 Reclassification 35
2.2.2 Aggregation 38
2.3 Agricultural Lands in the Year 2000 (Agland2000) 41
2.4 Harvested Area and Yields of 175 Crops (175Crops2000) 44

Chapter 3 Analysis **48**
3.1 Comparison of Aggregate Areas 49
3.2 NLCD Offsets 54
3.3 Fusion of Adjusted MLCT and Agland2000 58
3.4 Disaggregation of PEEL Crop Fractions According to 175Crops2000 60

Chapter 4 Conclusions **73**

References **75**

Appendix: Source Code **77**

List of Tables

1.1 Summary of global LULC data sets 2

2.1 Reclassification of MLCT/IGBP to PEEL (adapted from Friedl (2002)) 10
2.2 Reclassification of NLCD to PEEL (adapted from Homer et al. (2004)) 35
2.3 Crop sub-classes for simplifying 175Crops2000 (adapted from Monfreda et al. (2008)) 45

3.1 Total Acreages by Map and Cover 50
3.2 RMSE, MLCT vs. Agland2000 crop 52
3.3 RMSE, MLCT vs. Agland2000 crop with NLCD offsets 57
3.4 Effect of NLCD offsets on total acreages, $A_{min} = 0.5$ 57
3.5 RMSE of PEEL vs. Agland2000 59
3.6 PEEL acreages, $A_{min} = 0.5$ 59

List of Figures

2.1 MLCT primary cover reclassified detail . . . 11
2.2 MLCT secondary cover reclassified detail . . . 12
2.3 MLCT primary cover classification confidence detail . . . 13
2.4 MLCT primary covers shown separately, detail . . . 14
2.5 MLCT secondary covers shown separately, detail . . . 15
2.6 MLCT primary cover reclassified . . . 18
2.7 MLCT secondary cover reclassified . . . 19
2.8 MLCT primary cover classification confidence . . . 20
2.9 MLCT primary covers shown separately . . . 21
2.10 MLCT primary covers shown separately (cont.) . . . 22
2.11 MLCT secondary covers shown separately . . . 23
2.12 MLCT secondary covers shown separately (cont.) . . . 24
2.13 Sub-pixel fractions at original resolution for $A_{min} = 0.5$ . . . 25
2.14 Sub-pixel fractions at original resolution for $A_{min} = 1$ . . . 26
2.15 Aggregated sub-pixel fractions for $A_{min} = 1$ . . . 27
2.16 Aggregated sub-pixel fractions for $A_{min} = 0.5$ . . . 28
2.17 Difference of aggregated sub-pixel fractions, positive when $f(A_{min} = 0.5)$ is greater . . . 29
2.18 Aggregated cover fractions after mosaic decomposition, $A_{min} = 1.0$ . . . 31
2.19 Aggregated cover fractions after mosaic decomposition, $A_{min} = 0.5$ . . . 32
2.20 Differences of sub-pixel fractions after mosaic decomposition, positive when $f(A_{min} = 0.5)$ is greater . . . 33
2.21 NLCD reclassified . . . 36
2.22 NLCD covers shown separately, detail . . . 37
2.23 NLCD aggregated cover fractions, detail area . . . 38
2.24 NLCD aggregated cover fractions . . . 39
2.25 NLCD aggregated cover fractions (cont.) . . . 40
2.26 Agland2000 distribution in detail area . . . 42
2.27 Agland2000 distribution in cUSA study area . . . 43
2.28 175Crops2000 category maps . . . 46
2.29 175Crops2000 category maps (cont.) . . . 47

3.1 Total Acreages by Map and Cover . . . 51
3.2 Difference between MLCT (no mosaic, $A_{min} = 1.0$) and Agland2000 crop . . . 52
3.3 Difference between MLCT (no mosaic, $A_{min} = 0.5$) and Agland2000 crop . . . 53
3.4 Hexbin plot of MLCT crop ($A_{min} = 1.0$, no mosaic) versus Agland2000 cropland . . . 54
3.5 Hexbin plot of MLCT crop ($A_{min} = 0.5$, no mosaic) versus Agland2000 cropland . . . 55
3.6 NLCD offsets . . . 61
3.7 NLCD offsets (cont.) . . . 62
3.8 Correlation matrix of NLCD offsets . . . 63
3.9 Total offsets calculated from NLCD . . . 64
3.10 Total acreages after NLCD adjustment . . . 65
3.11 Hexbin plot of MLCT adjusted crop versus Agland2000 cropland . . . 66
3.12 Final PEEL maps . . . 67
3.13 Final PEEL cover maps (cont.) . . . 68
3.14 Hexbin plot of PEEL crop versus Agland2000 crop . . . 69
3.15 Conflicts between NLCD offsets and Agland2000 . . . 70
3.16 Normalized fractions for crop sub-classes . . . 71
3.17 Normalized fractions for crop sub-classes (cont.) . . . 72

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)
AVHRR	Advanced Very High Resolution Radiometer
CIM-EARTH	Community Integrated Model of Economic and Resource Trajectories for Humankind
cUSA	conterminous (contiguous) Unites States of America, the "lower 48"
GADM	Global Administrative Areas, `http://www.gadm.org/`
GIAM	Global Irrigated Areas Map
GIS	Geographic Information Systems
GMRCA	Global Map of Rainfed Crop Areas
GLC2000	Global Land Cover 2000 (European Commission, Joint Research Centre, 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 (LP DAAC, 2008)
NLCD	National Land-Cover Database, 2001 (Homer et al., 2004)
PEEL	Partial Equilibrium Economic Land use model
PLSS	Public Land Survey System
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
[0,1]	interval from 0 to 1, inclusive of 0 and 1; $0 \leq x \leq 1$
[0,1)	interval from 0 to 1, inclusive of 0 but but not 1, $0 \leq x < 1$

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 et al., 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 feasible, as we will demonstrate 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 yet still reflects 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 readily applicable to theoretical modeling.

Data set	Reference	Sensor	Resolution	Time Span
UMD Global Land Cover 1998	Hansen et al. (2000)	AVHRR	1km	1981 – 1994 (composite)
Global Land Cover 2000	European Commission, Joint Research Centre (2003); Bartholomé and Belward (2005)	SPOT	1km	Nov 1999 – Dec 2000 (composite)
National Land Cover Database (NLCD)	Homer et al. (2004, 2007)	Landsat	30 m	2001
MODIS Land Cover Type v005	LP DAAC (2008); Friedl et al. (2010)	MODIS (Aqua & Terra)	500m	2001 – 2008 (annual time series)

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 Agri-

cultural 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, adaptability, 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 it 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 homogeneous 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 sub-pixel 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. 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 auto-correlation among neighbors to prevent unrealistic divergences in development patterns among grid cell neighborhoods, for example.

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 multi-spectral 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 gaging 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 performed 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 research 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 commercial, 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 in-line 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 principles 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 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 disciplines 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 `R` environment under which this analysis was performed is as follows:

```
R version 2.13.0 (2011-04-13)
Platform: x86_64-pc-linux-gnu (64-bit)

attached base packages:
 [1] splines   grid      tools
 [4] stats     graphics  grDevices
 [7] utils     datasets  methods
[10] base

other attached packages:
 [1] hexbin_1.26.0
 [2] lattice_0.19-26
 [3] doMC_1.2.1
 [4] multicore_0.1-3
 [5] foreach_1.3.0
 [6] codetools_0.2-8
 [7] iterators_1.0.3
 [8] RColorBrewer_1.0-2
 [9] Hmisc_3.8-3
[10] survival_2.36-9
[11] xtable_1.5-6
[12] ggplot2_0.8.9
[13] proto_0.3-9.2
```

```
[14] reshape_0.8.4
[15] plyr_1.5.2
[16] rgdal_0.6-33
[17] raster_1.8-12
[18] sp_0.9-80

loaded via a namespace (and not attached):
[1] cluster_1.13.3 digest_0.4.2
```

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/thesis` and/or `http://www.cimearth.org/` or by request to `mailto:nbest@ci.uchicago.edu` or `mailto:nbest@alum.mit.edu`.

Chapter 2

Data Sets and Methodology

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 definitions 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 an 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 cover 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 simplified classification system. This general case is also compared to simpler cases in aggregating NLCD and considering only the primary classification of MLCT where the secondary class and classification confidence variables are ignored. 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 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 diversity 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 the author, being his birthplace.

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

MLCT/IGBP		PEEL	
0	water	0	water
1 2 3 4 5	evergreen needleleaf forest deciduous needleleaf forest evergreen broadleaf forest deciduous broadleaf forest mixed forests	1	forest
6 7 8	closed shrublands open shrublands woody savannas	2	shrub
9 10	savannas grasslands	3	open
11	permanent wetlands	4	wetland
12	croplands	5	crop
13	urban	6	urban
14	cropland / natural vegetation mosaics	7	mosaic
15 16	permanent snow and ice barren or sparsely vegetated	8	barren

Table 2.1: Reclassification of MLCT/IGBP to PEEL (adapted from Friedl (2002))

Table 2.1 shows the mapping of the IGBP classes used in the original MLCT data to the simplified classification designed for the PEEL model. Collapsing the five forest categories into a single class was an easy call. Making a distinction between woody savannas and savannas and assigning them to the shrub and

open classes respectively is supported by the IGBP class definitions (Friedl, 2002) due to the overlap in the forest canopy cover for those IGBP classes. This makes sense in the context of the PEEL model because the ecological roles, potential uses, and conversion costs of the two savanna types are subjectively dissimilar. Areas of permanent snow and ice were combined with barren or sparsely vegetated areas, which would include deserts, to form the PEEL barren class based on their shared characteristics of low population density and relatively low intensity of economic activity.

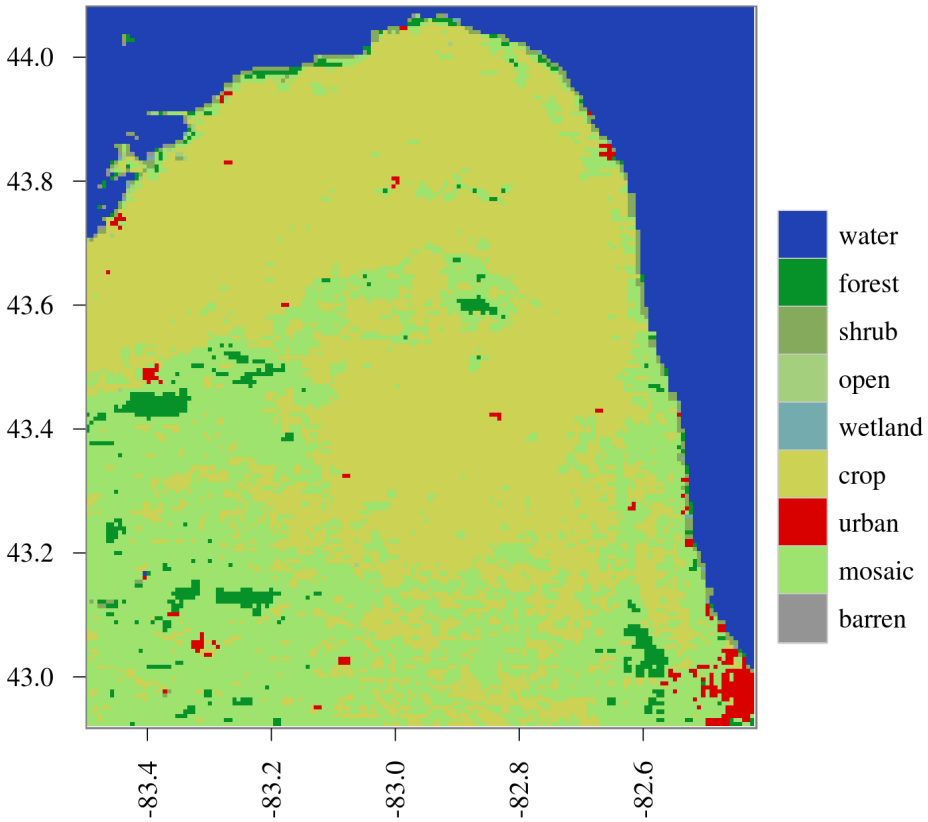

Figure 2.1: MLCT primary cover reclassified detail

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, Michigan and Sarnia, Ontario is visible in the southeast corner.

In Figure 2.2 we notice that areas in the northern and central sections of the map that were classified as

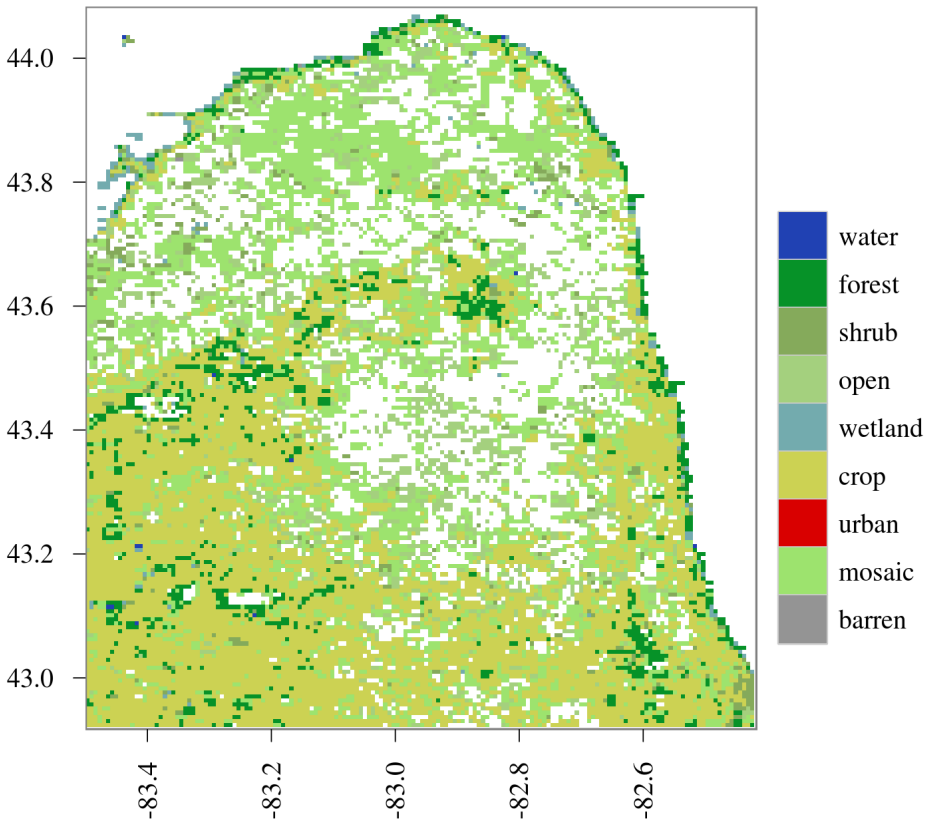

Figure 2.2: MLCT secondary cover reclassified detail

crop in the primary layer have null values in the secondary class. It is apparent that where a secondary class is given 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 indicates 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 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,

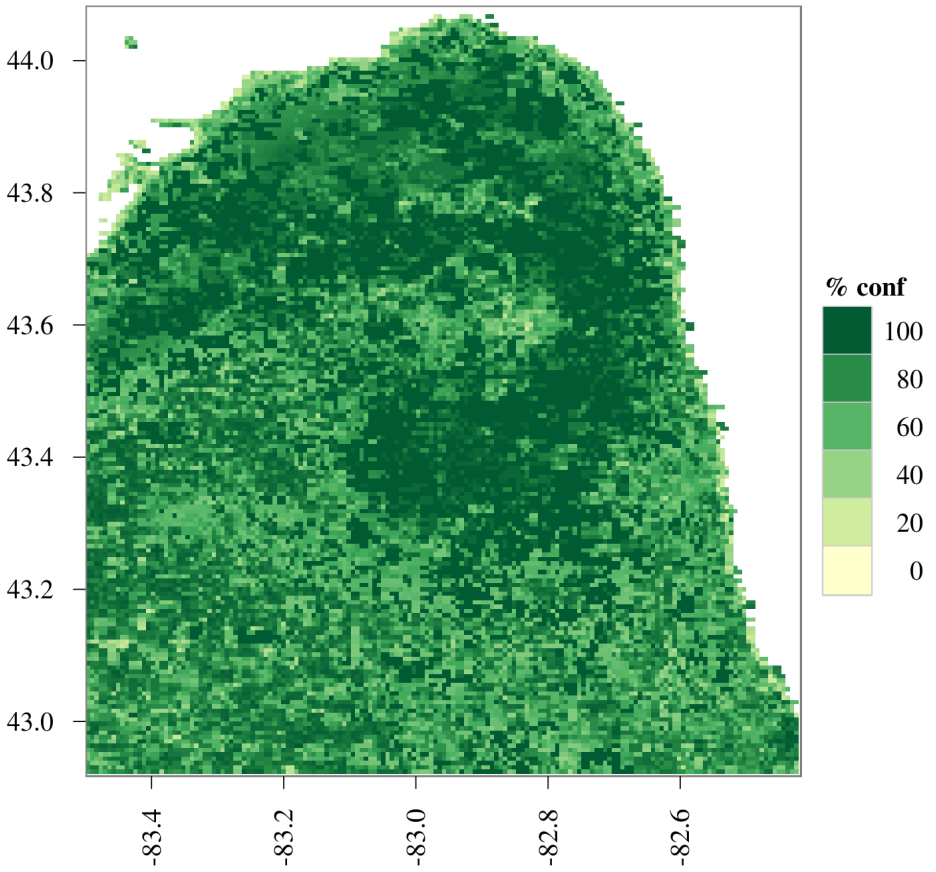

Figure 2.3: MLCT primary cover classification confidence detail

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.

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 2.11 have done. From these

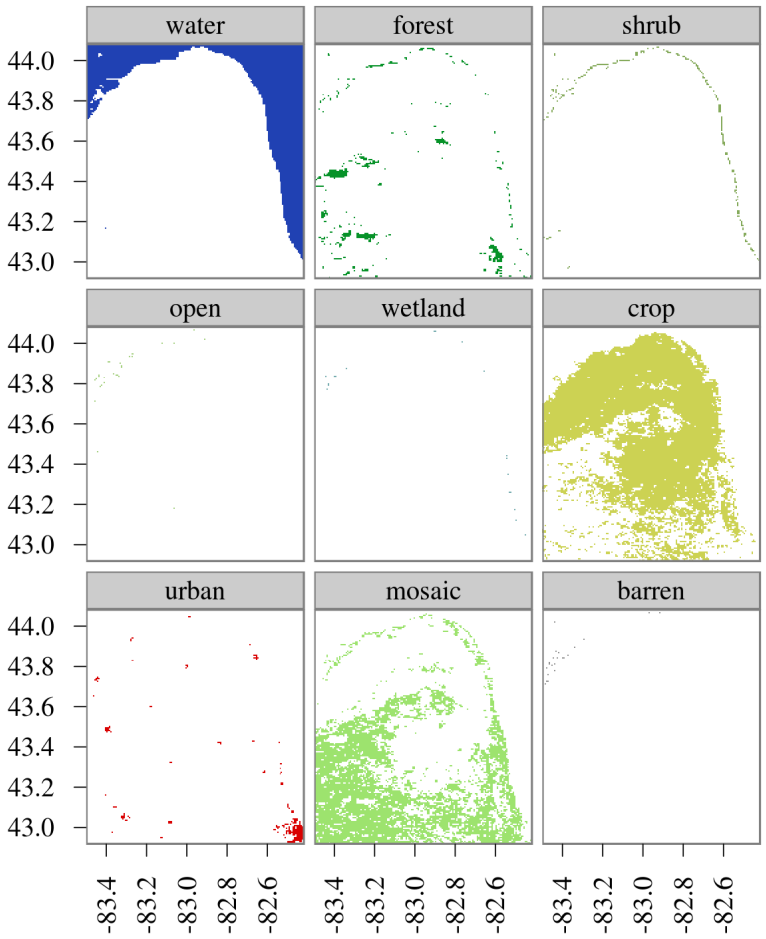

Figure 2.4: MLCT primary covers shown separately, detail

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.

2.1.2 Aggregation

MLCT has a nominal resolution of 500m which roughly equates to 15″ at the equator 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 of this aggregation will be a function of the 400 original MLCT pixels

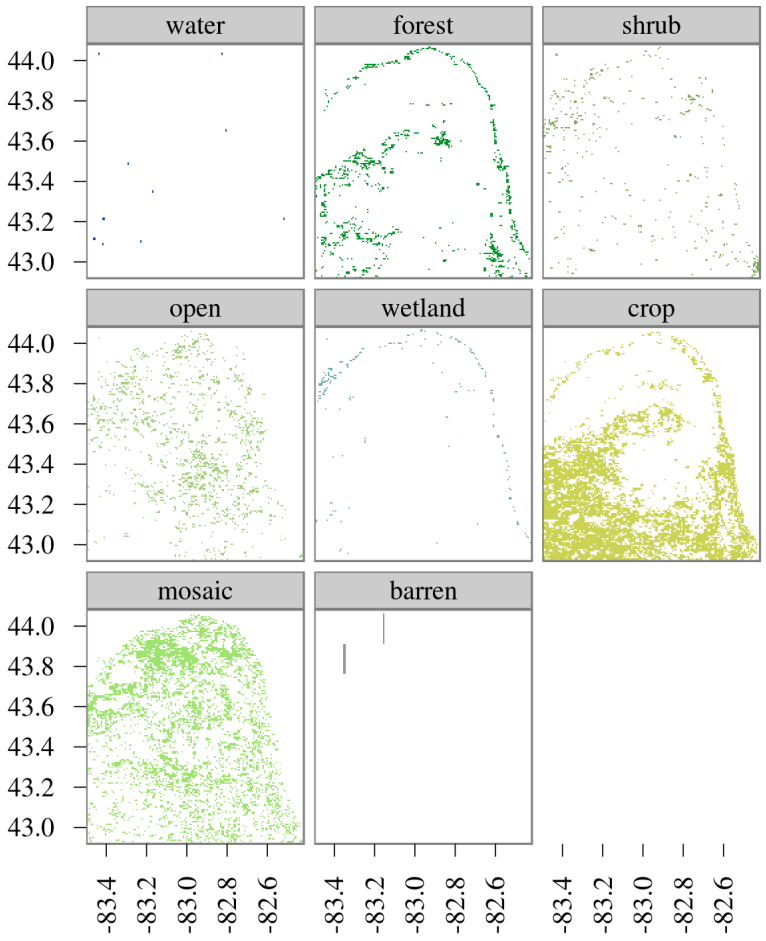

Figure 2.5: MLCT secondary covers shown separately, detail

within its footprint. The data set 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. We are not as concerned about per-pixel classification error as users working at the MLCT's native resolution might be, which is the original motivation given for providing the secondary classification. Because we are aggregating the data up to $5'$ resolution there is no expectation that the sub-pixel fractions at full resolution are spatially specific, but in the aggregate our characterization of each grid cell's composition will be nuanced by this additional information. By definition the primary class covers no more than 60% of a given pixel x (Friedl, 2002). For the purposes of this analysis we assume that the relationship between classification confidence and the sub-pixel fraction of the primary class is a linear, monotonically increasing function. Thus, for the

primary and secondary cover types in a pixel:

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

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

where A_{min} represents the minimum area fraction to be assigned to the primary class given $c(x) = 0$. It stands to reason that a given class would have to comprise more than 50% of the pixel area in order to be considered primary, therefore $A_{min} = 0.5$ affords maximum consideration to the secondary class in this scheme. Simplifying the equations by substituting this value gives:

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

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

Instances of $c < 0.20$ are rare, so generally the primary class will be assigned more than 60% of the MLCT pixel area. In the analysis that follows we will compare the product of these assumptions with the case of $A_{min} = 1.0$ which gives zero consideration to the secondary class.

Applying these formulas results in a map for each cover type where the pixel 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 2.13 shows the result of calculating $A_p + A_s$ for each individual class.

By way of comparison we also consider the trivial case of setting $A_{min} = 1$ which indicates that the secondary cover is ignored altogether and the primary cover is taken to represent 100% of the pixel area. Figure 2.14 shows these differences. 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 $5'$ 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 resolution. 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.

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 2.17. 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 similarly for mosaic in the south. The relative suitability of these choices for A_{min} is discussed in chapter 3.

Figure 2.17 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, 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.

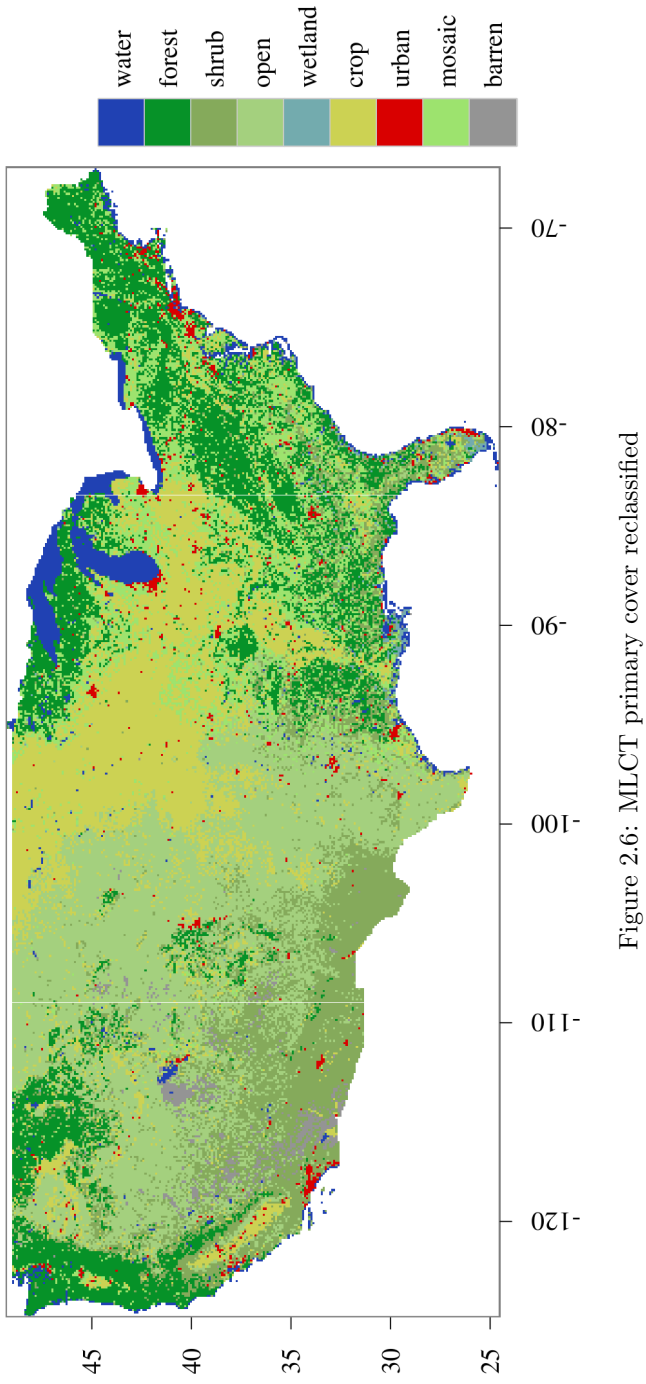

Figure 2.6: MLCT primary cover reclassified

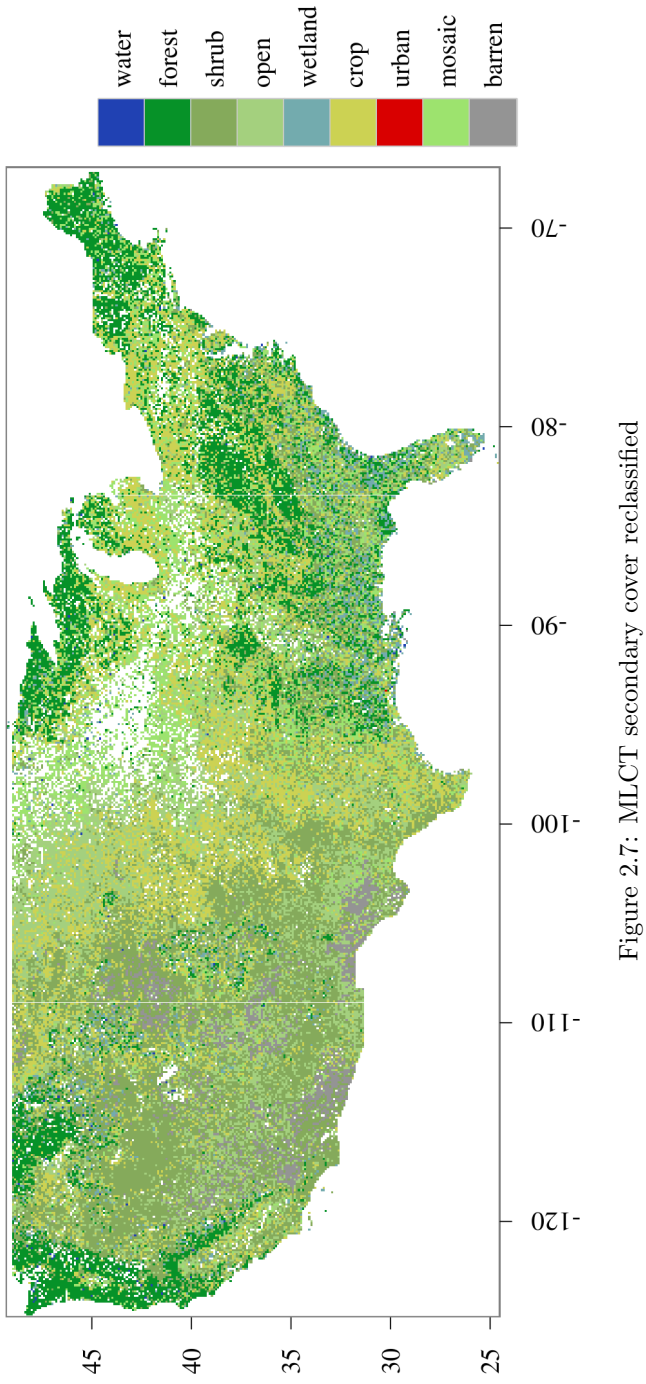

Figure 2.7: MLCT secondary cover reclassified

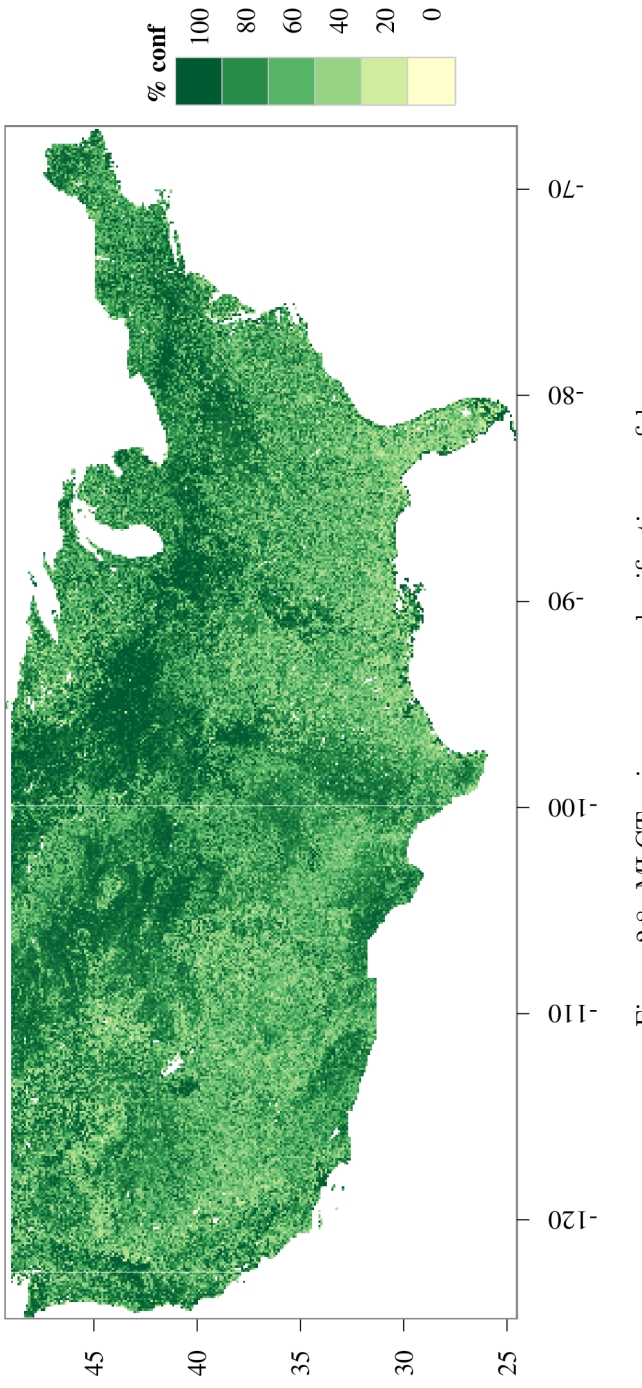

Figure 2.8: MLCT primary cover classification confidence

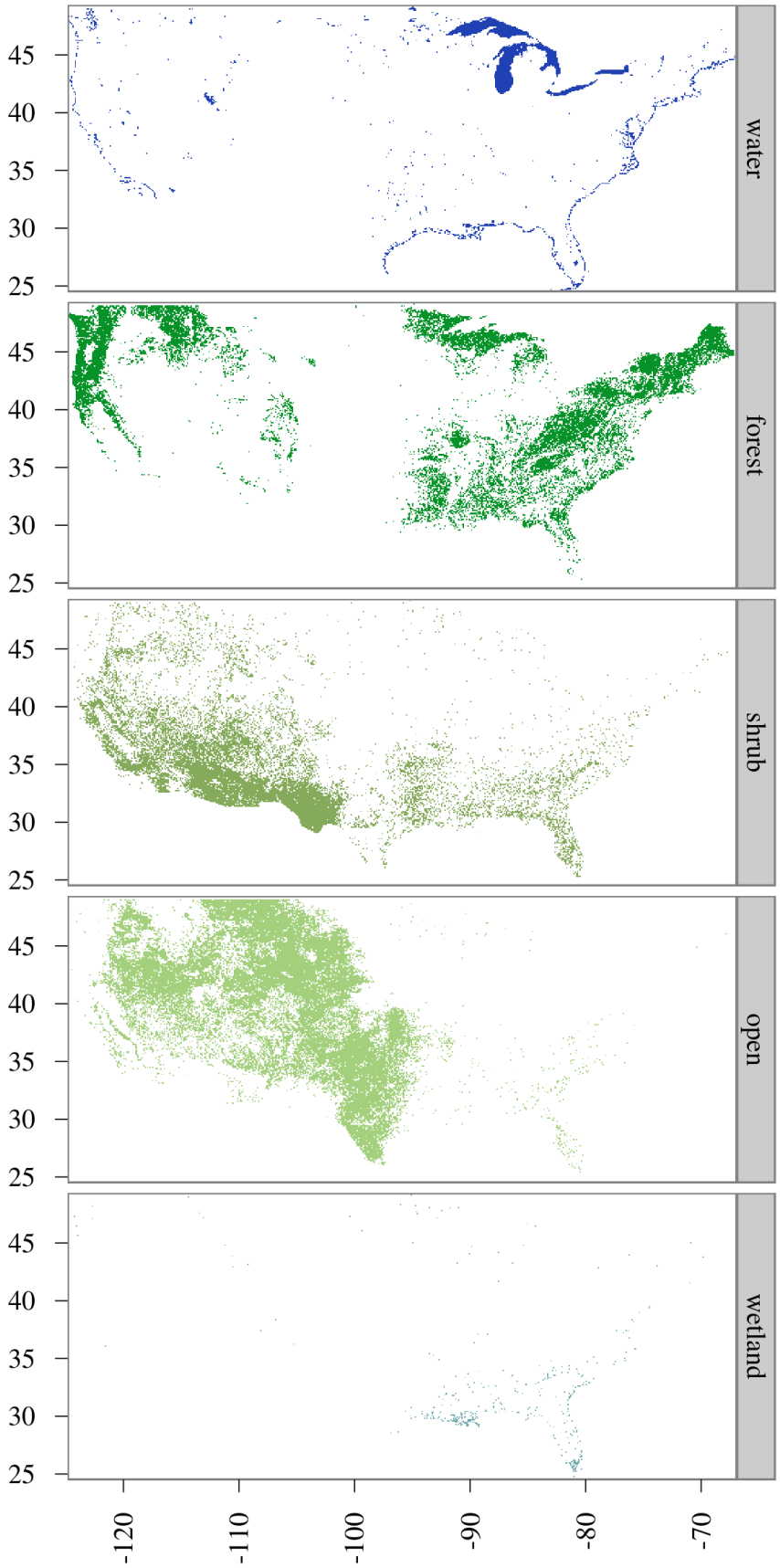

Figure 2.9: MLCT primary covers shown separately

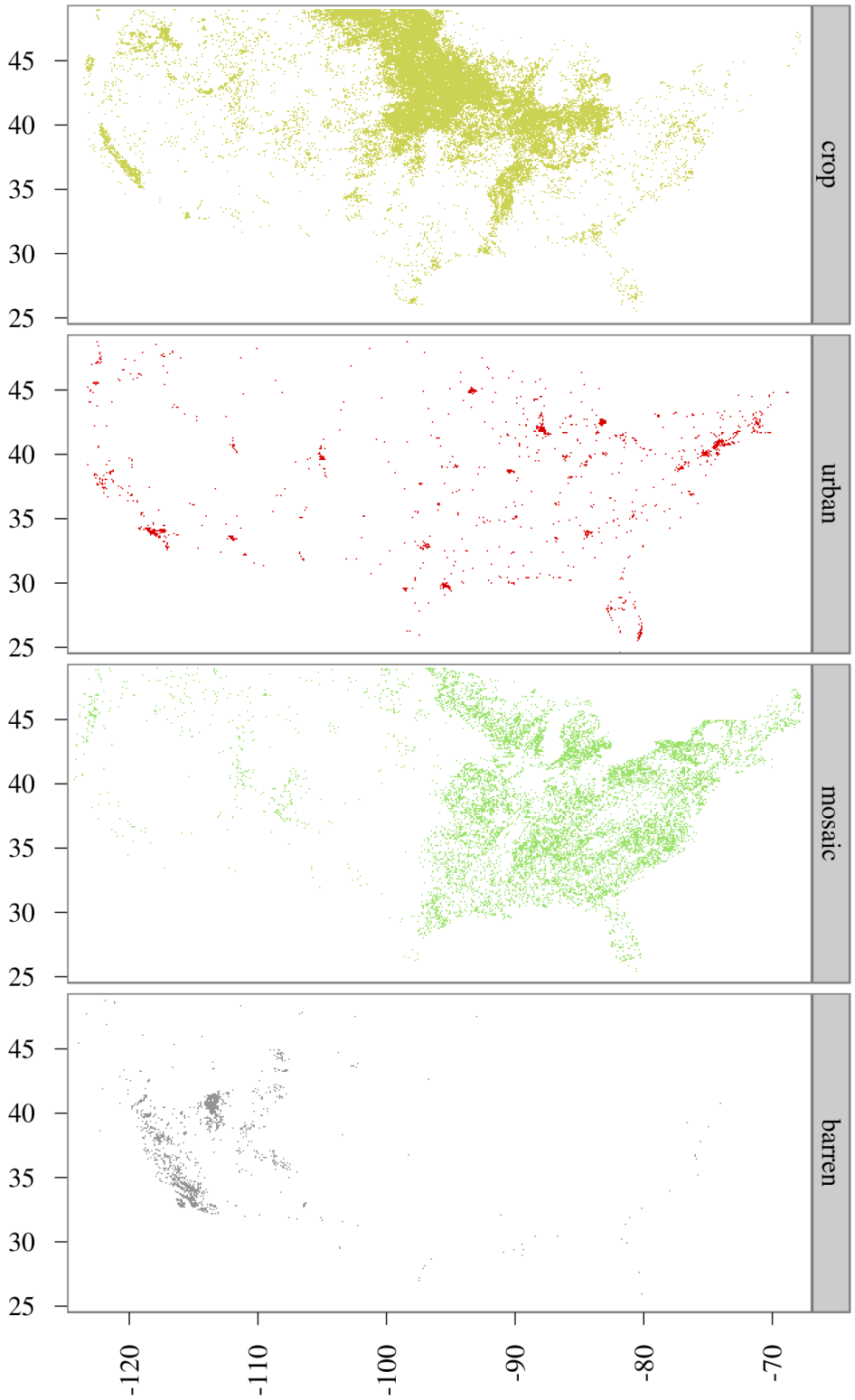

Figure 2.10: MLCT primary covers shown separately (cont.)

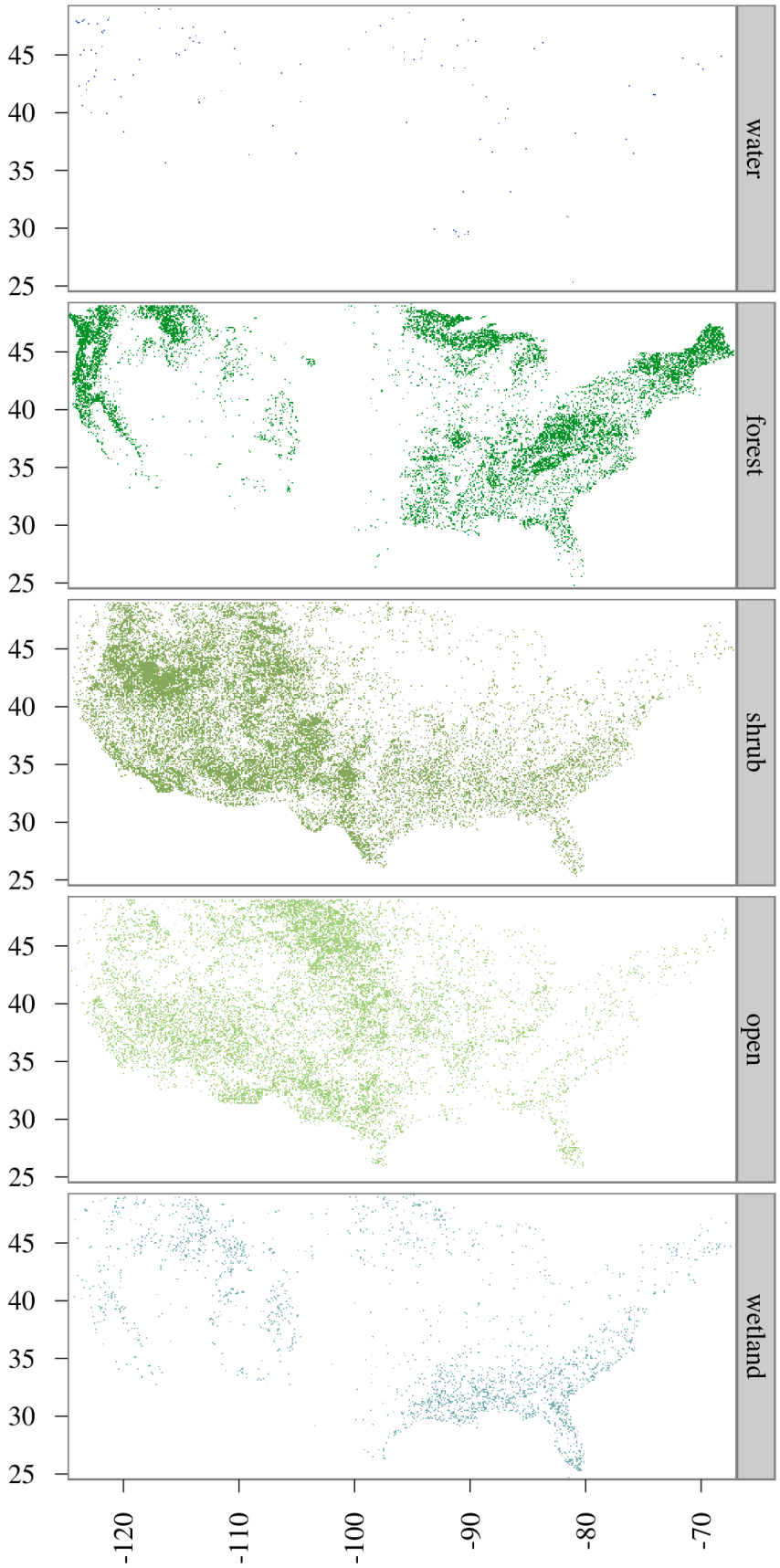

Figure 2.11: MLCT secondary covers shown separately

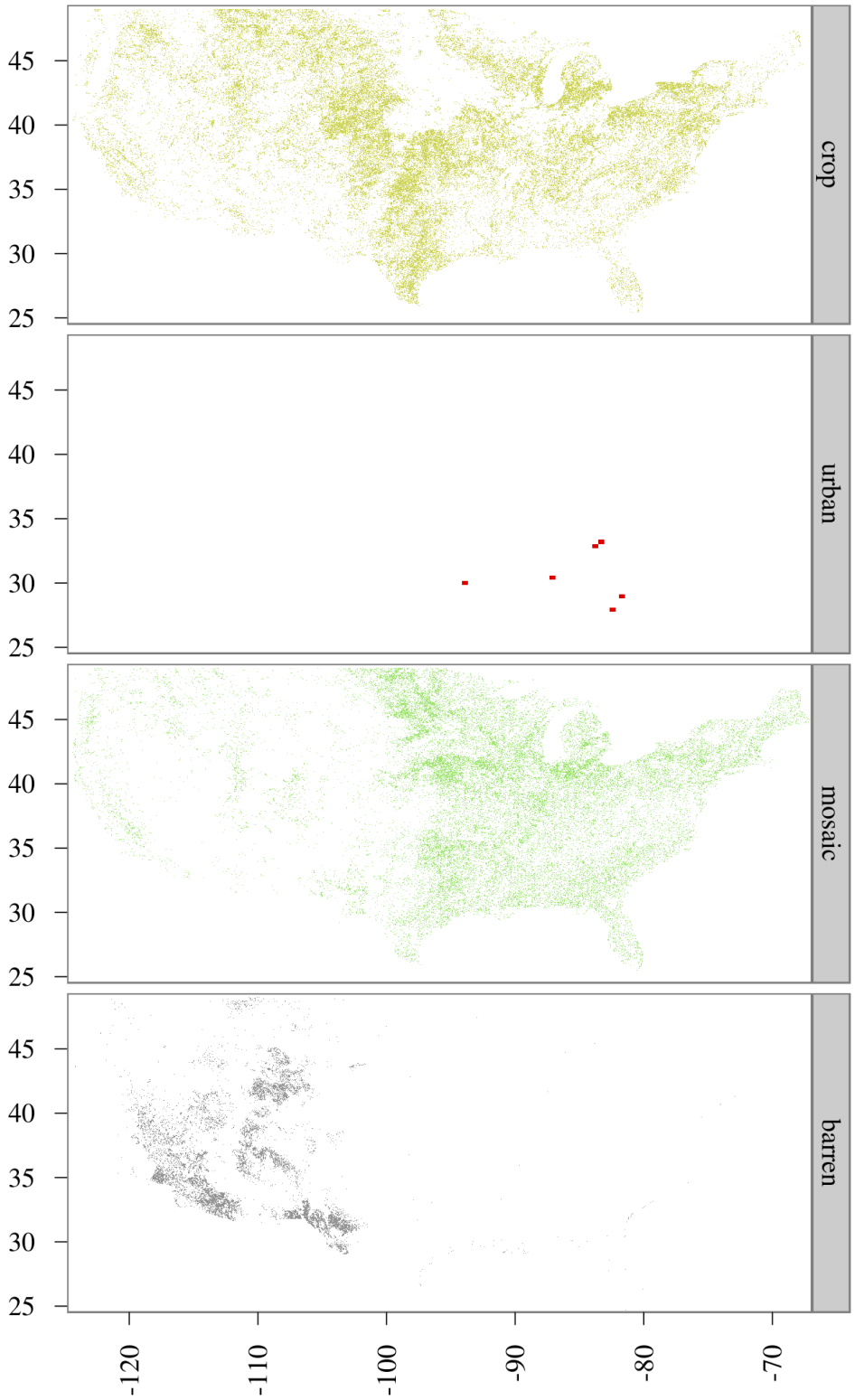

Figure 2.12: MLCT secondary covers shown separately (cont.)

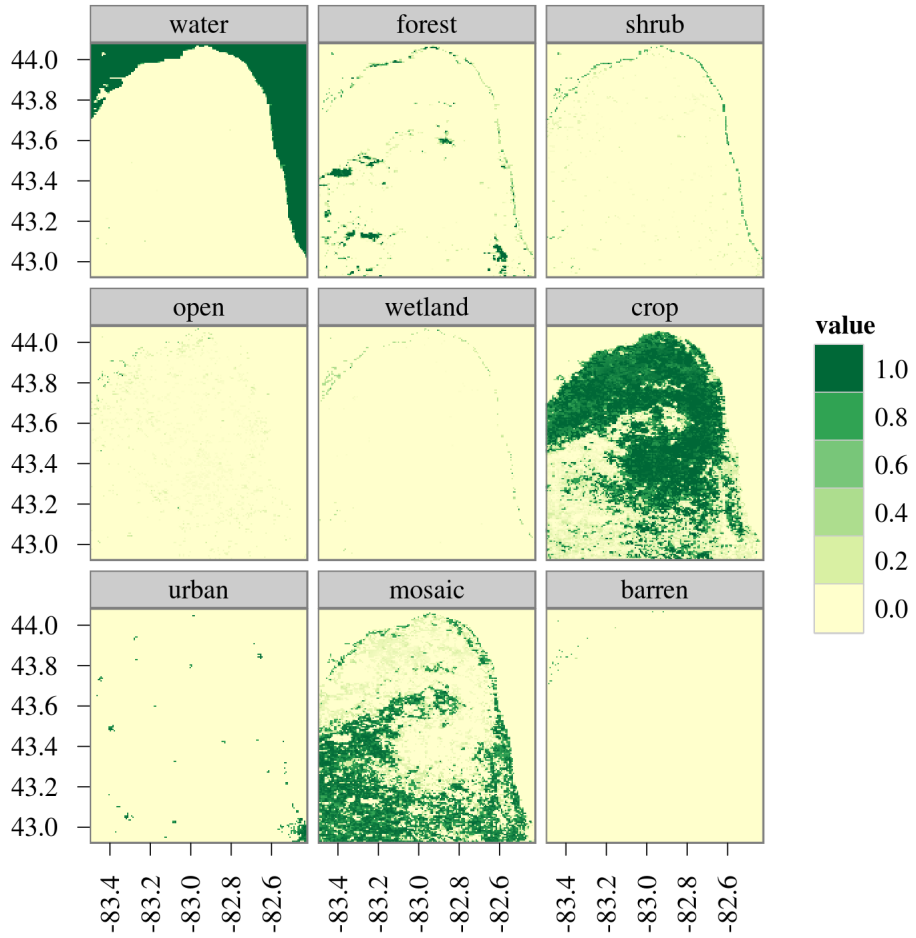

Figure 2.13: Sub-pixel fractions at original resolution for $A_{min} = 0.5$

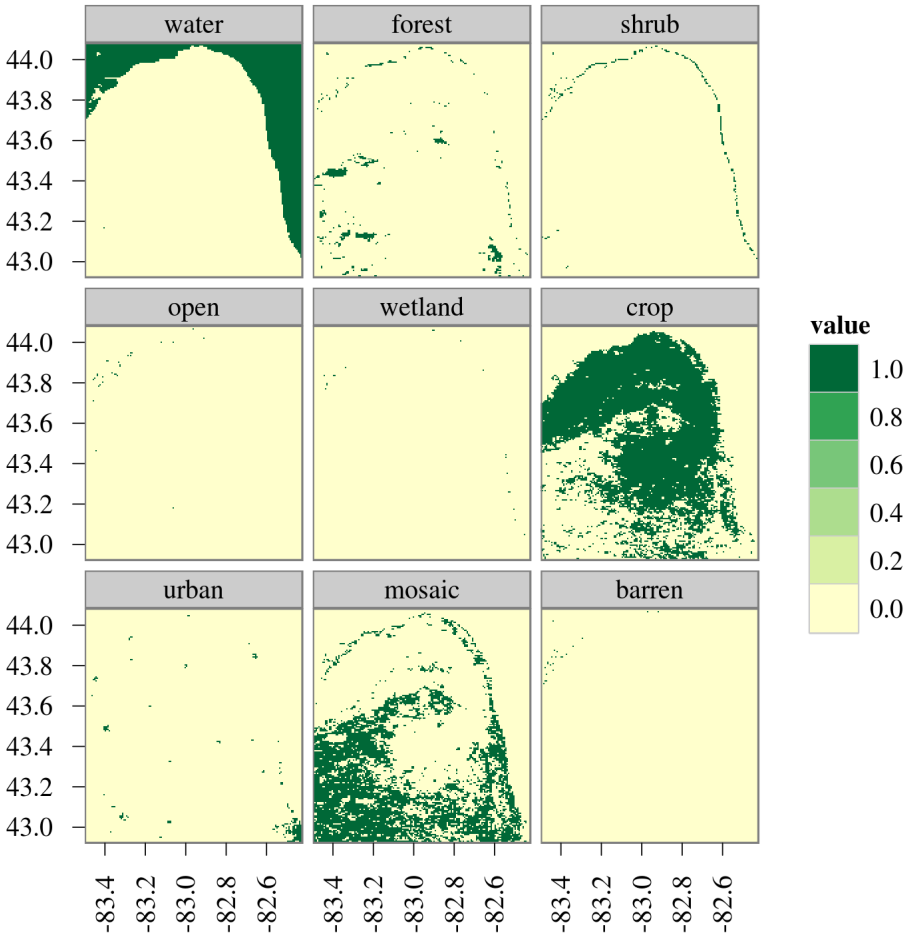

Figure 2.14: Sub-pixel fractions at original resolution for $A_{min} = 1$

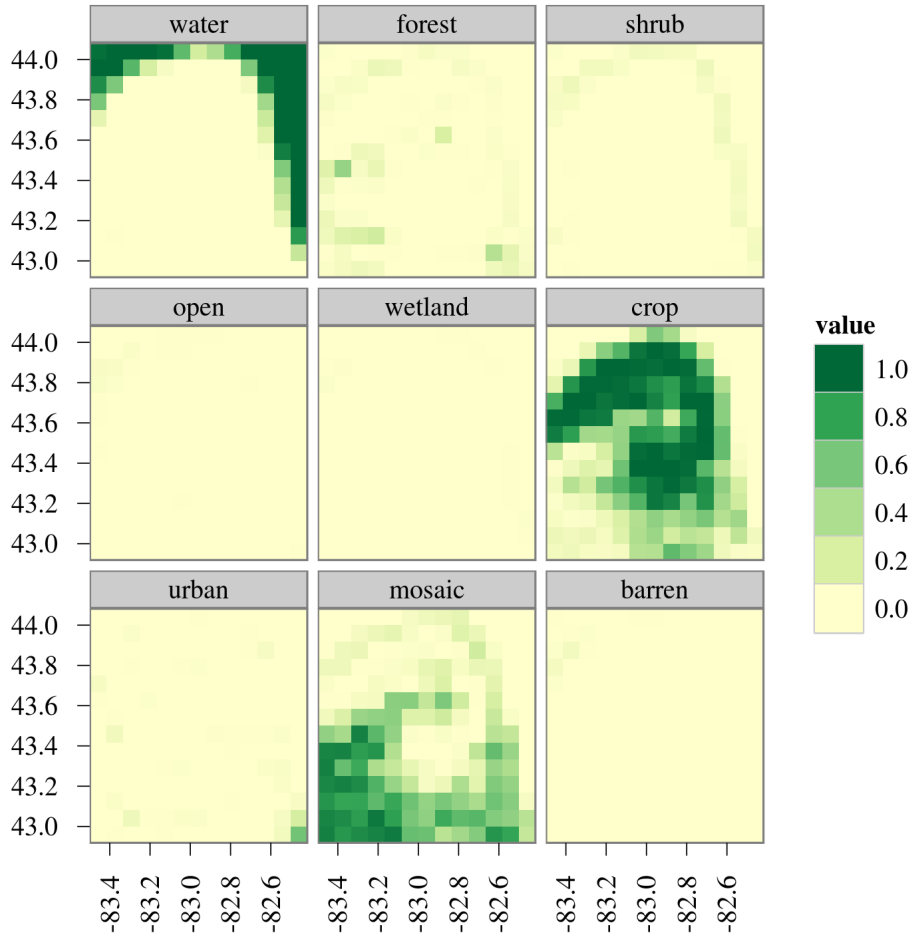

Figure 2.15: Aggregated sub-pixel fractions for $A_{min} = 1$

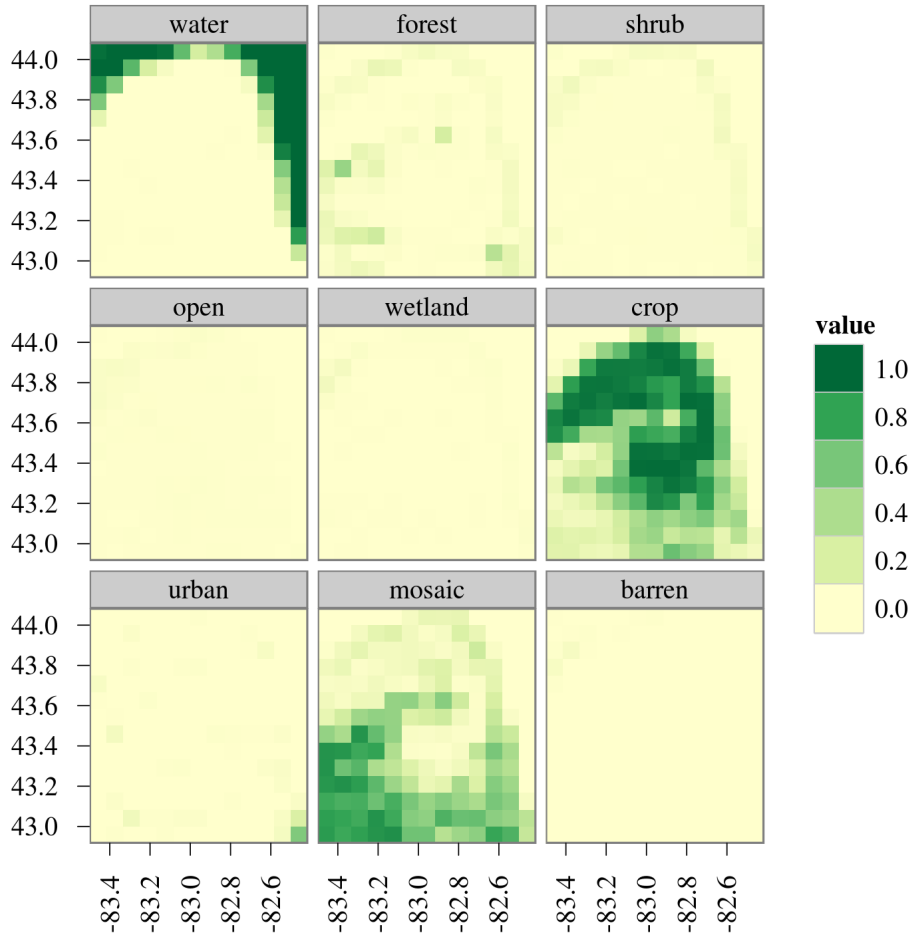

Figure 2.16: Aggregated sub-pixel fractions for $A_{min} = 0.5$

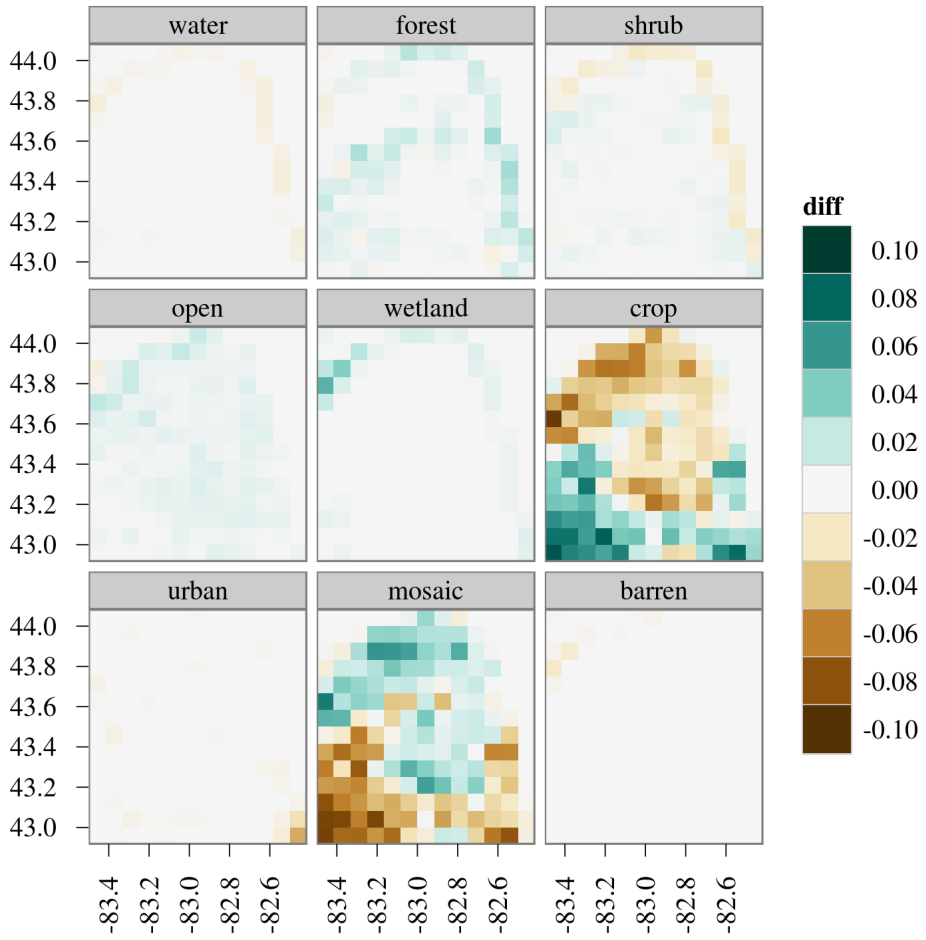

Figure 2.17: Difference of aggregated sub-pixel fractions, positive when $f(A_{min} = 0.5)$ is greater

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). 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:

1. 50% of mosaic area is assigned to the crop class.
2. The other 50% is a blend of forest, open, and shrub in proportion to the expression of those classes in the same $5'$ cell.
3. In the absence of any natural classes in the $5'$ cell 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 definition 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 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 2.19 and Figure 2.18 show the effect of decomposing the mosaic class in this fashion for A_{min} values of 0.5 and 1.0 respectively. Figure 2.20 shows the difference between the two maps that result from the mosaic decomposition process in the same manner as the previous difference map. Values are positive when $A_{min} = 0.5$ results in a greater value. This difference map shows that the effect of the choice of A_{min} on the crop class is much less pronounced than in the previous step, rather it is the natural cover classes

that also make up the mosaic class that exhibit shifts in relative composition. The open class is virtually nonexistent in the primary classification but has a strong component in the secondary classification (see Figure 2.1 and Figure 2.2), which changes the balance of the natural covers when calculating the mosaic decomposition as described above.

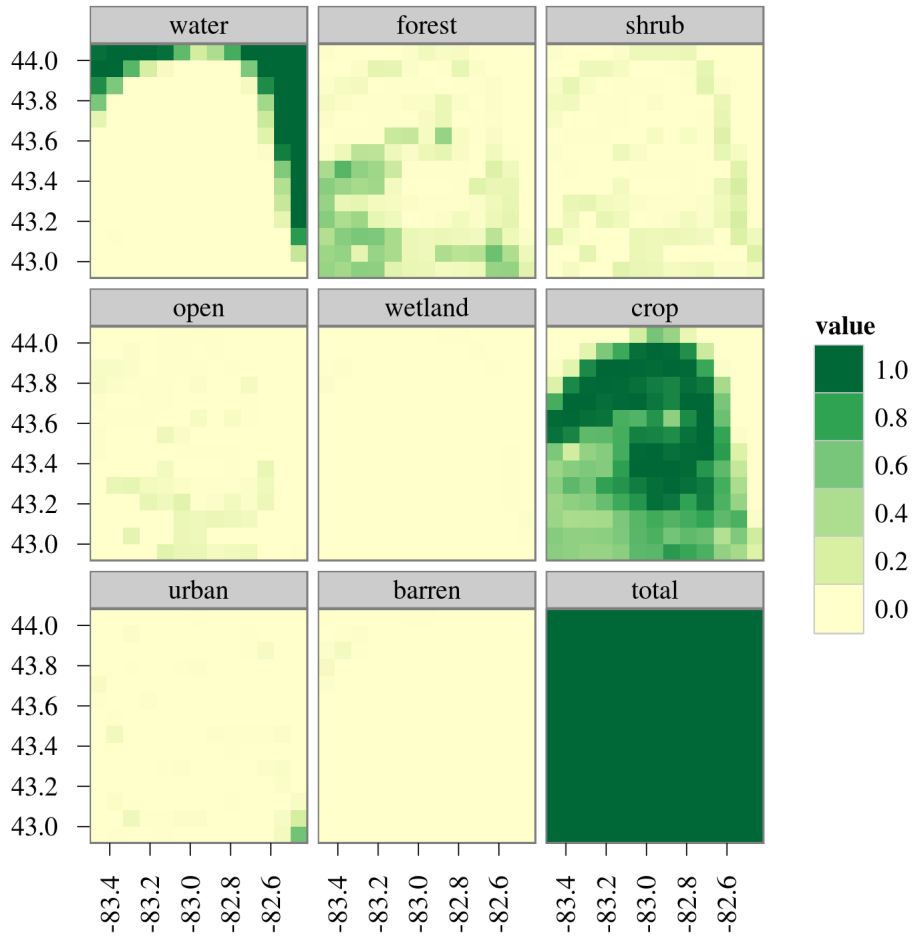

Figure 2.18: Aggregated cover fractions after mosaic decomposition, $A_{min} = 1.0$

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.

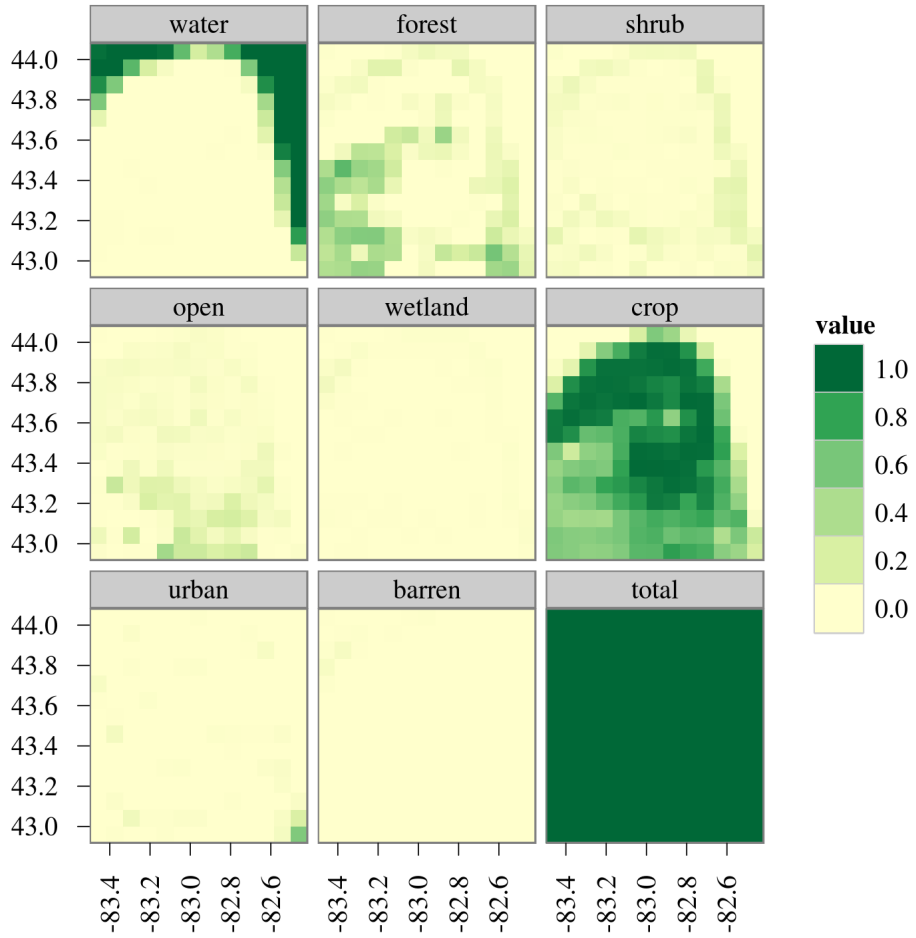

Figure 2.19: Aggregated cover fractions after mosaic decomposition, $A_{min} = 0.5$

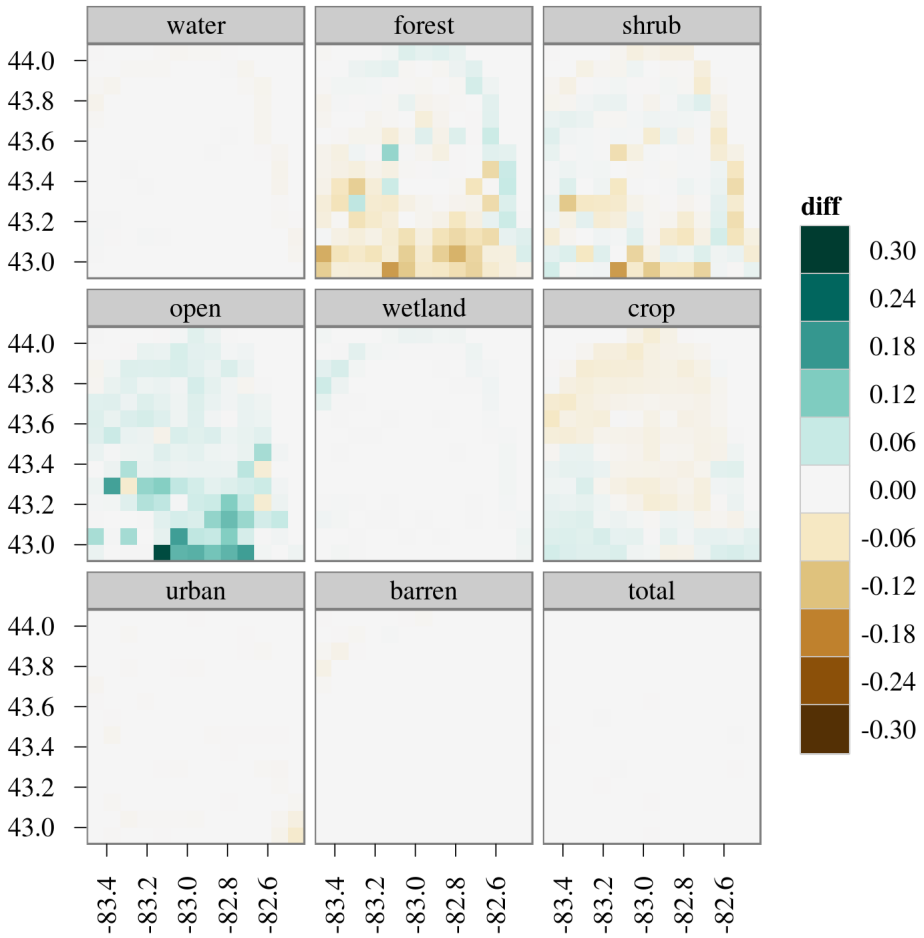

Figure 2.20: Differences of sub-pixel fractions after mosaic decomposition, positive when $f(A_{min} = 0.5)$ is greater

2.2 National Land-cover Database 2001 (NLCD)

The NLCD gives a higher-resolution (30m) snapshot of LULC across the cUSA study area, plus Alaska, Hawaii and Puerto Rico, circa 2001. Because NLCD's classification was informed by ancillary data sets such as population density, buffered roads, and the National Wetland Inventory (Homer et al., 2004) reclassifying and aggregating this data to $5'$ 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. Although it is unclear from Homer et al. (2004) what ancillary data was applied in what constituent mapping zones of the NLCD we accept its representation of these fine details to be the best available data. This will compensate for MLCT's bias against these finely 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.

2.2.1 Reclassification

NLCD		PEEL	
11	water	0	water
98	palustrine aquatic bed*		
99	estuarine aquatic bed*		
41	deciduous forest	1	forest
42	evergreen forest		
43	mixed forest		
52	shrub/scrub	2	shrub
94	estuarine scrub/shrub wetland*		
71	grassland / herbaceous	3	open
81	pasture / hay		
90	woody wetlands	4	wetland
91	palustrine forested wetland*		
92	palustrine scrub/shrub wetland*		
93	estuarine forested wetland*		
95	emergent herbaceous wetlands		
96	palustrine emergent wetland (persistent)*		
97	palustrine scrub/shrub wetland*		
82	cultivated crops	5	crop
21	developed, open space	6	urban
22	developed, low intensity		
23	developed, medium intensity		
24	developed, high intensity		
	(no equivalent)	7	mosaic
12	perennial ice/snow	8	barren
31	barren land		
32	unconsolidated shore		

* Indicates coastal classes

Table 2.2: Reclassification of NLCD to PEEL (adapted from Homer et al. (2004))

In contrast to the reclassification table for MLCT shown in Table 2.1, the reclassification for NLCD given in Table 2.2 is somewhat more complicated. Although NLCD has fewer forest classes they are equally unambiguous. On the other hand NLCD has some special coastal classes that we chose to distribute among PEEL classes that we decided were most similar qualitatively. Those classes are indicated with an asterisk in Table 2.2. NLCD features four classes of developed land that we are equating with the PEEL urban class to represent developed areas of all densities.

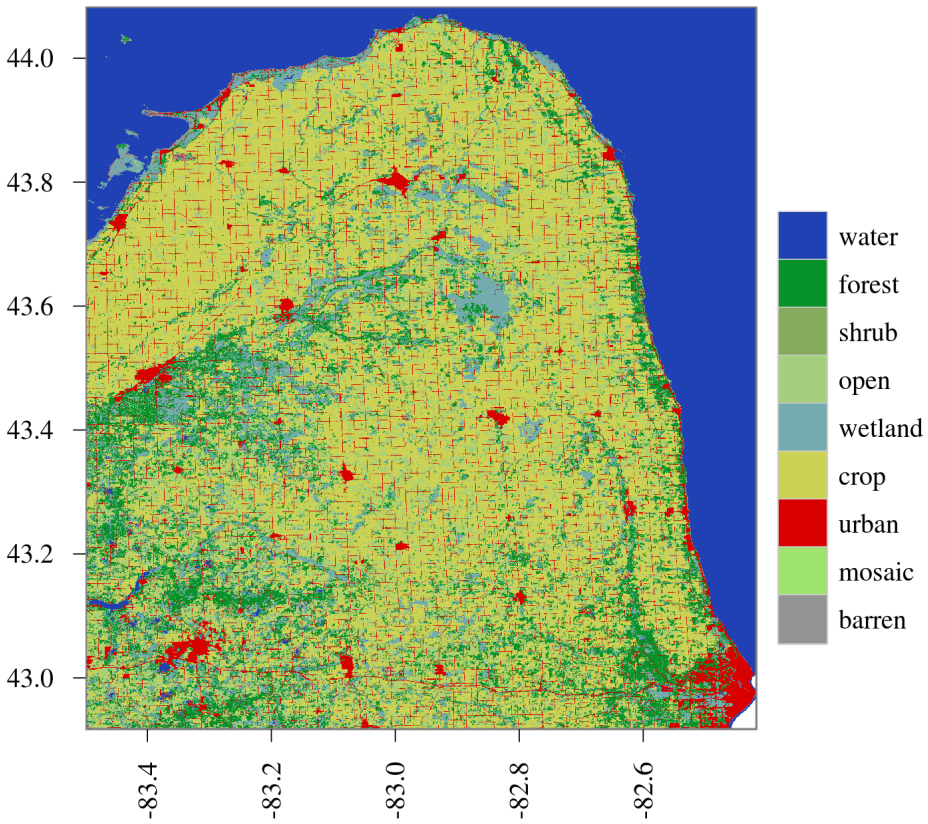

Figure 2.21: NLCD reclassified

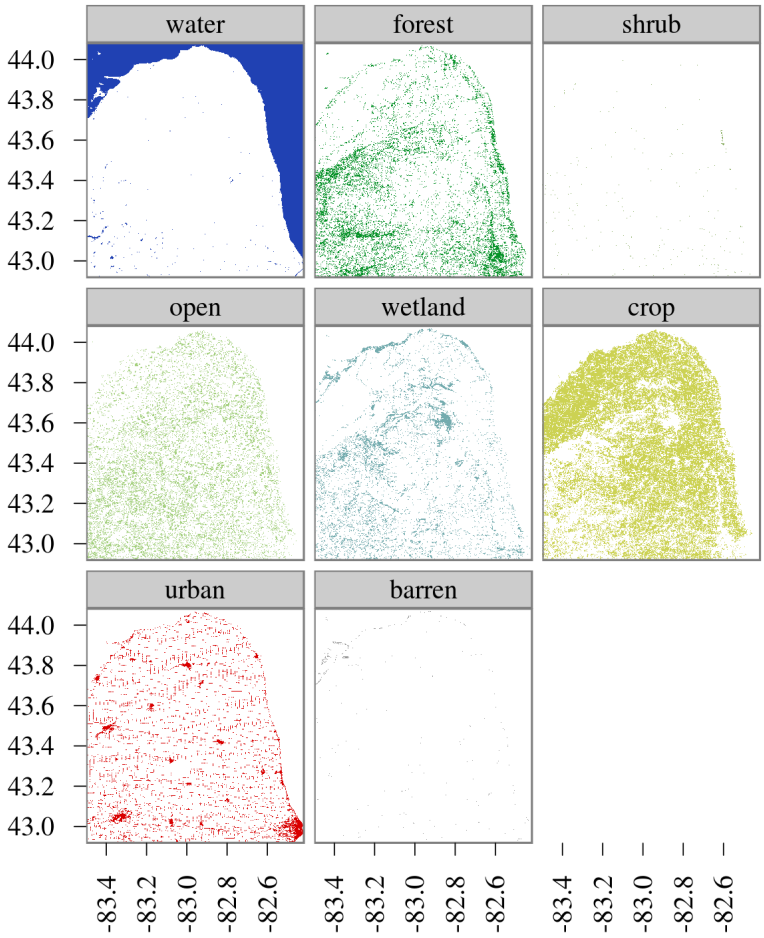

Figure 2.22: NLCD covers shown separately, detail

2.2.2 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.

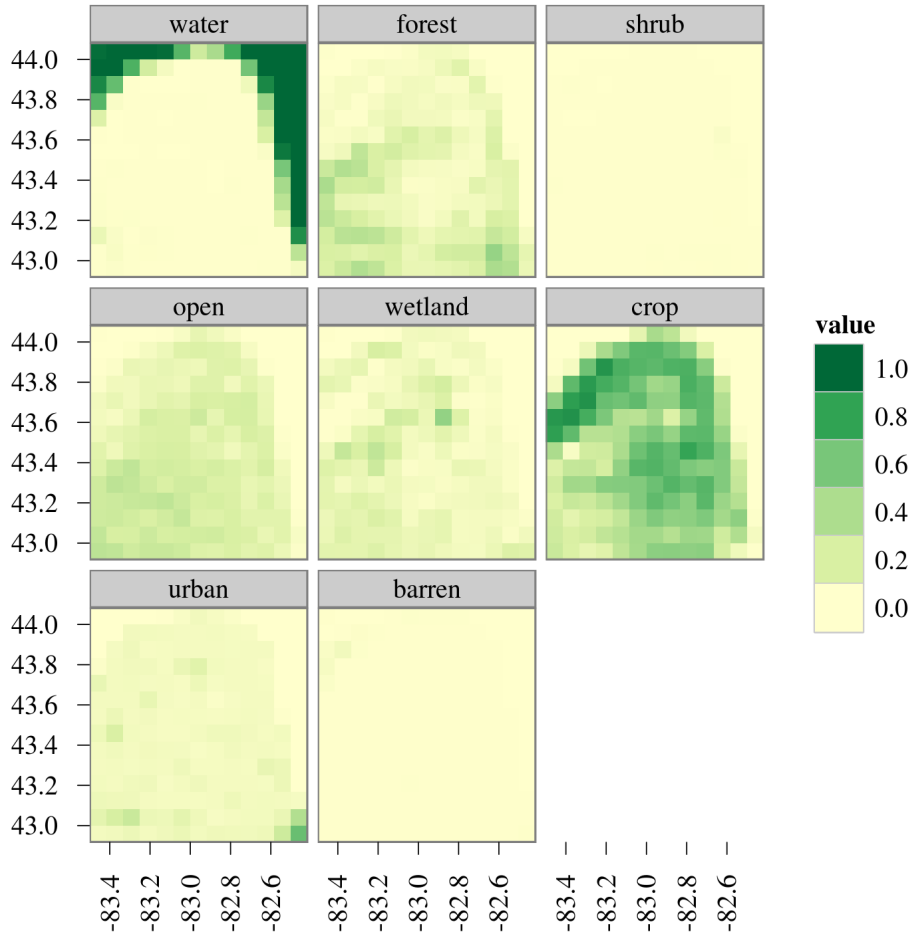

Figure 2.23: NLCD aggregated cover fractions, detail area

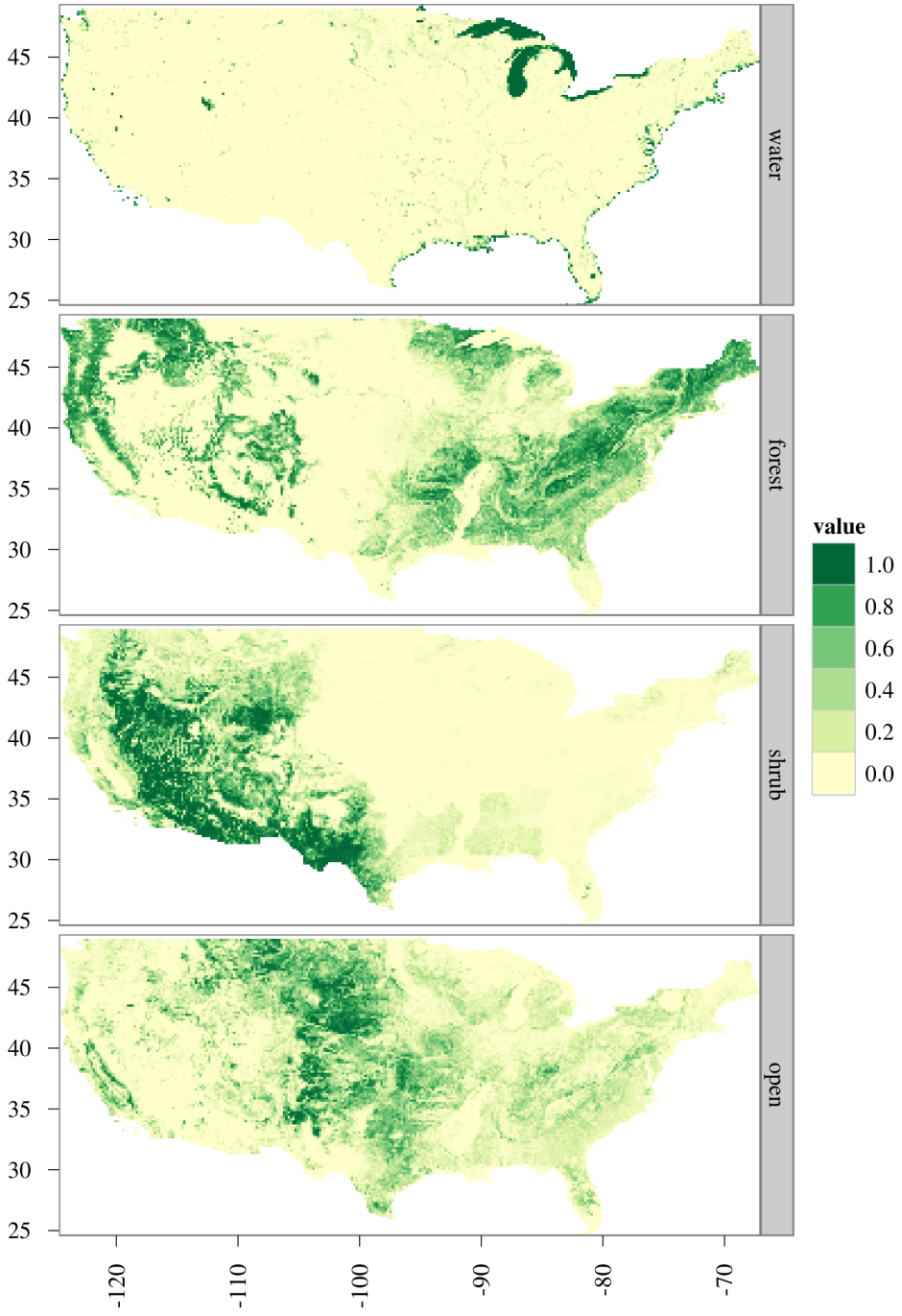

Figure 2.24: NLCD aggregated cover fractions

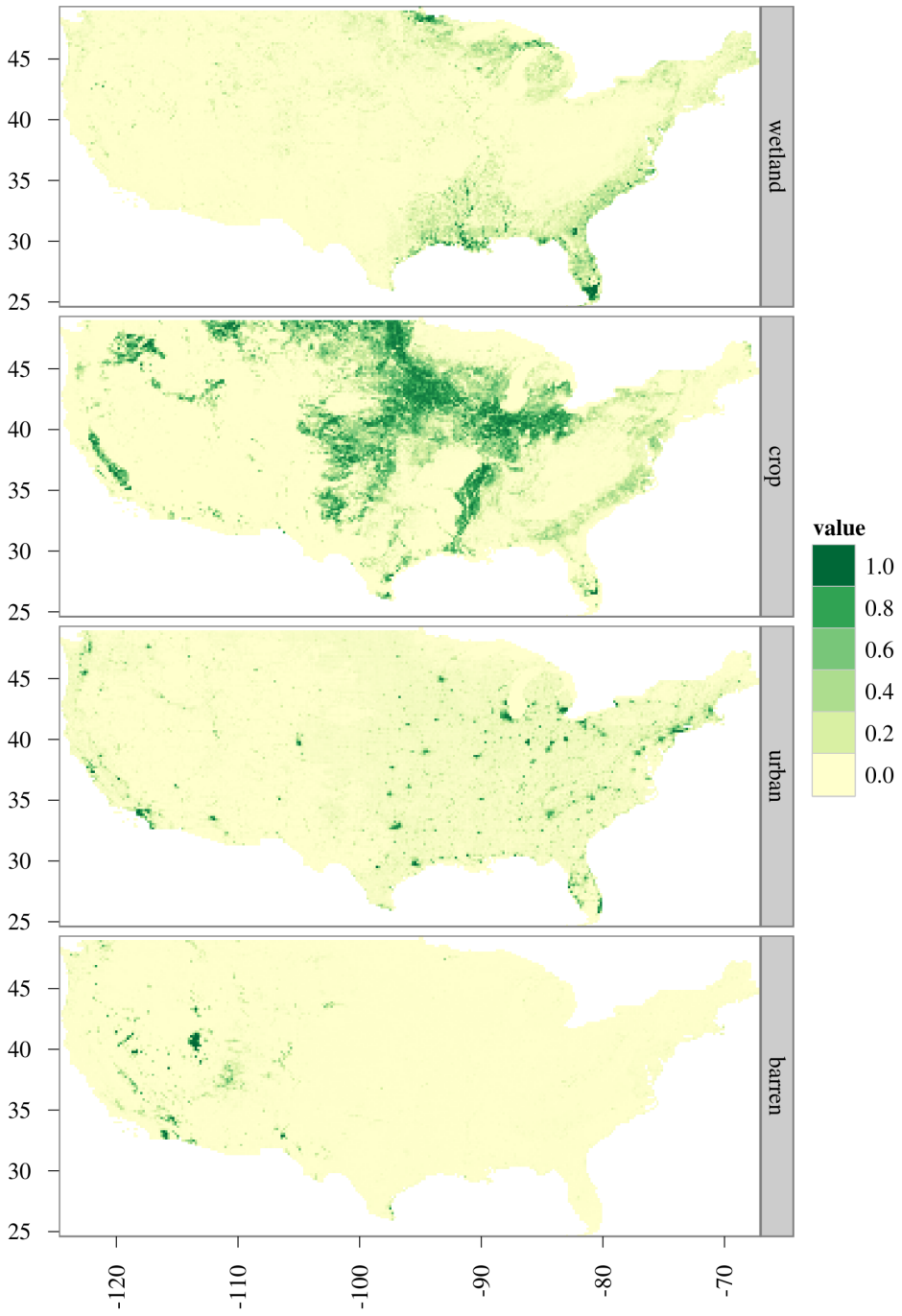

Figure 2.25: NLCD aggregated cover fractions (cont.)

2.3 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 arable land and permanent crops compiled at national or sub-national levels according to availability of such data at or near the turn of the last century. It uses two classified LULC data sets derived from remote sensing data as inputs, an older version of the MLCT (known as BU-MODIS) and the GLC2000 data set mentioned in section 1.1. Its allocation of cropland and pastures is constrained by a mask based on climatic criteria in order to avoid misallocation at higher latitudes beyond our study area. The "pasture" class in this data set likely has much in common with the "open" class from MLCT but we are not employing that data in this analysis. It is important to note that the cropland aspect of 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.

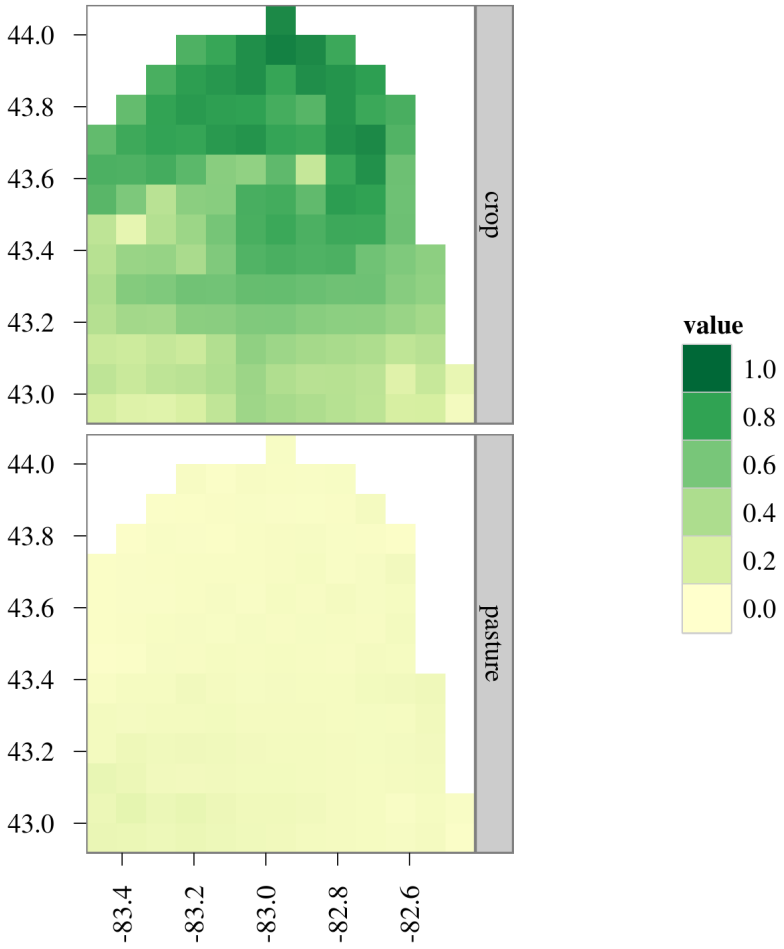

Figure 2.26: Agland2000 distribution in detail area

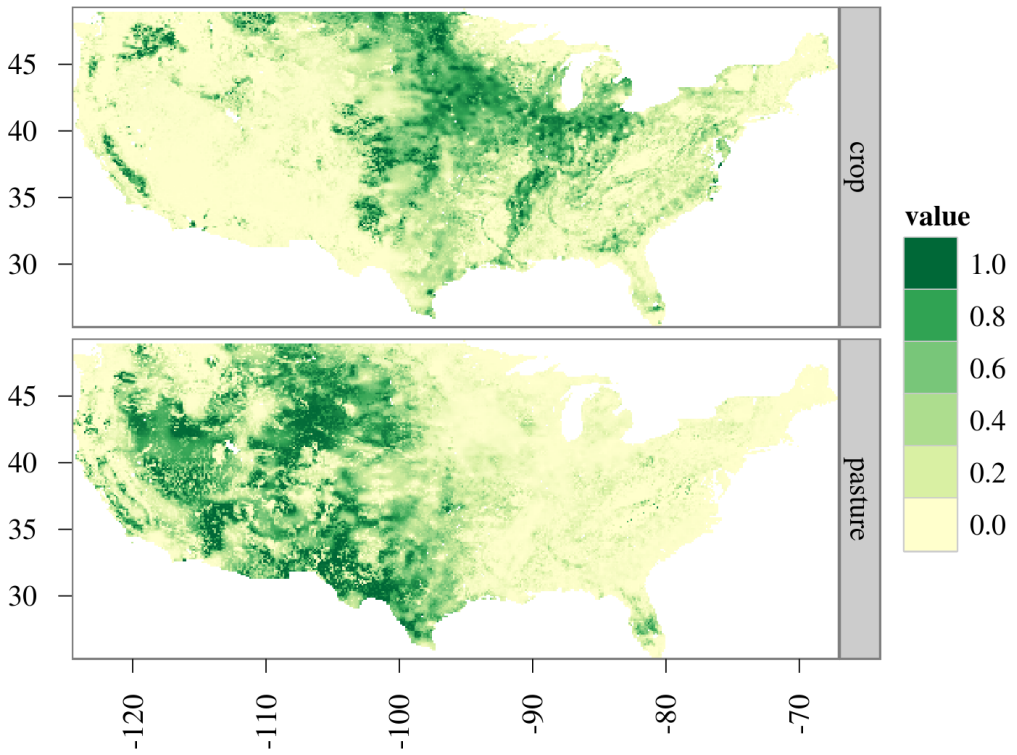

Figure 2.27: Agland2000 distribution in cUSA study area

2.4 Harvested Area and Yields of 175 Crops (175Crops2000)

This data set, described in Monfreda et al. (2008), will provide the information needed to disaggregate the cropland area taken from Agland2000. It is not possible to use this data directly 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. Table 2.3 provides the details on how the 175 crops in the Monfreda et al. (2008) data set are collected into crop sub-classes for the purposes of the PEEL model.

Sub-class	Crops
maize	maize
wheat	wheat
rice	rice
other cereals	barley; buckwheat; canary seed; cereals nes; fonio; millet; mixed grain; oats; pop corn; quinoa; rye; sorghum; triticale
soybean	soybean
sugarcane	sugarcane
forage	alfalfa for forage; beets for fodder; cabbage for fodder; carrots for fodder; clover; forage products nes; grasses nes; green oilseeds fr fodder; legumes nes; maize for forage; mixed grasses and legumes; rye grass for forage; sorghum for forage; swedes for fodder; turnips for fodder; vegetables and roots for fodder
other field crops	anise, badian and fennel; artichokes; asparagus; bambara beans; beans, dry; beans, green; broad beans, dry; broad beans, green; cabbages; cantaloupes and other melons; carrots; cassava; castor beans; cauliflower; chickpeas; chicory roots; chillies and peppers, green; coir; cotton; cow peas, dry; cucumbers and gherkins; eggplants; fibre crops nes; flax fibre and tow; garlic; ginger; green corn (maize); groundnuts in shell; hemp fibre and tow; hempseed; jute; jute-like fibres; lentils; lettuce; linseed; lupins; melonseed; mushrooms; mustard seed; oilseeds nes; okra; onions and shallots, green; onions, dry; peas, dry; peas, green; peppermint; pigeon peas; pimento; poppy seed; potatoes; pulses nes; pumpkins, squash, gourds; rapeseed; roots and tubers nes; safflower seed; sesame seed; spinach; string beans; sugar beets; sugar crops nes; sunflower seed; sweet potatoes; taro; tobacco leaves; tomatoes; vegetables fresh nes; vetches; watermelons; yams; yautia
shrub crops	abaca (manila hemp); agave fibres nes; bananas; berries nes; blueberries; cocoa beans; coffee, green; cranberries; gooseberries; grapes; hops; mate; nutmeg and mace and cardamons; pepper; pineapples; plantains; pyrethrum, dried flowers; ramie; raspberries; sisal; strawberries; tea; vanilla
tree crops	almonds; apples; apricots; areca nuts (betel); avocados; brazil nuts; carobs; cashewapple; cashew nuts; cherries; chestnuts; cinnamon (canella); citrus fruit nes; cloves; coconuts; currants; dates; figs; fruit fresh nes; fruit tropical fresh nes; grapefruit and pomelos; hazelnuts (filberts); kapok fibre; kapokseed in shell; karite nuts (sheanuts); kiwi fruit; kolanuts; lemons and limes; mangoes; natural gums; natural rubber; nuts nes; oil palm fruit; olives; oranges; papayas; peaches and nectarines; pears; persimmons; pistachios; plums; quinces; sour cherries; spices nes; stone fruit nes, fresh; tang.mand.clement.satsma; tung nuts; walnuts

nes: "not elsewhere specified"

Table 2.3: Crop sub-classes for simplifying 175Crops2000 (adapted from Monfreda et al. (2008))

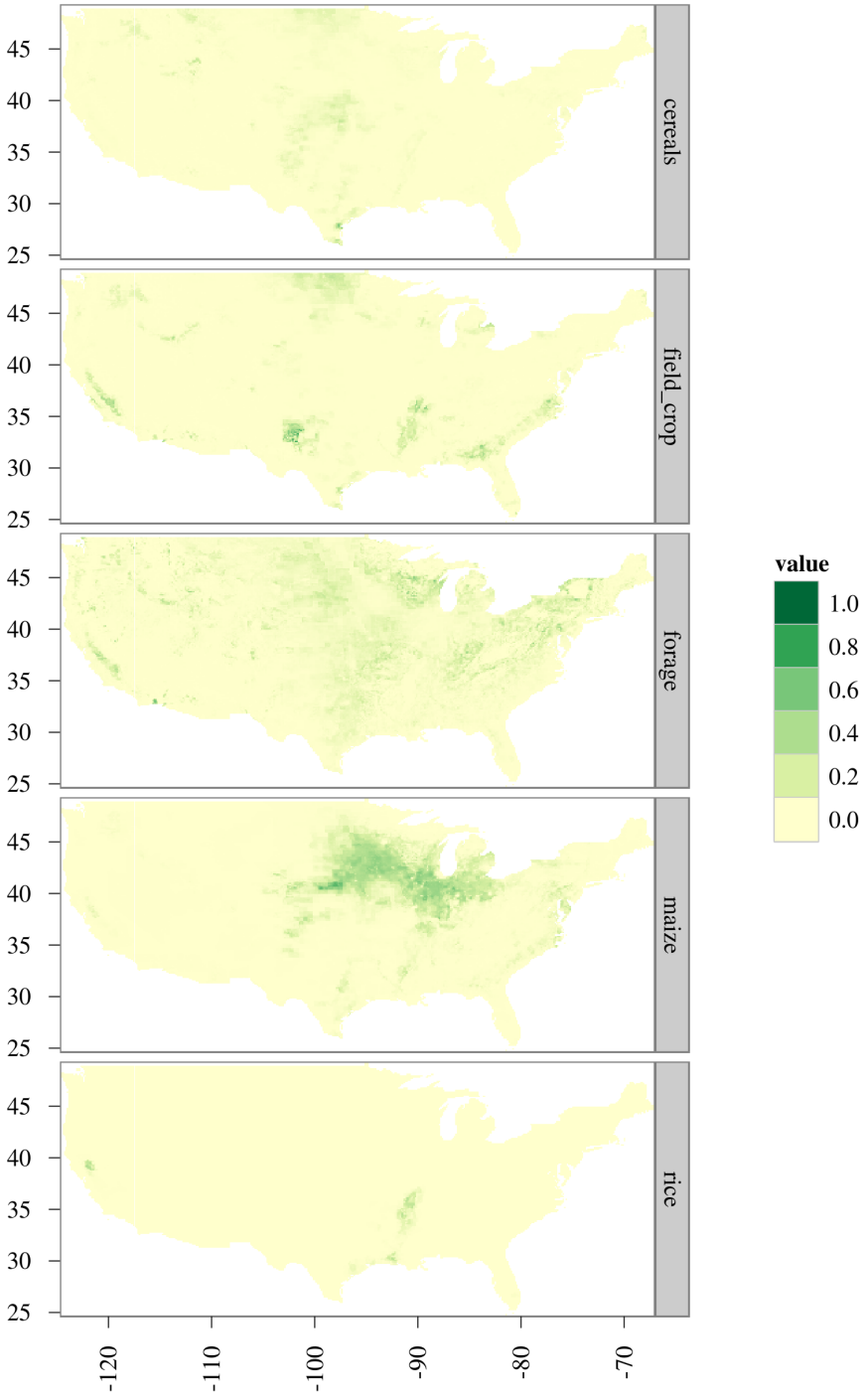

Figure 2.28: 175Crops2000 category maps

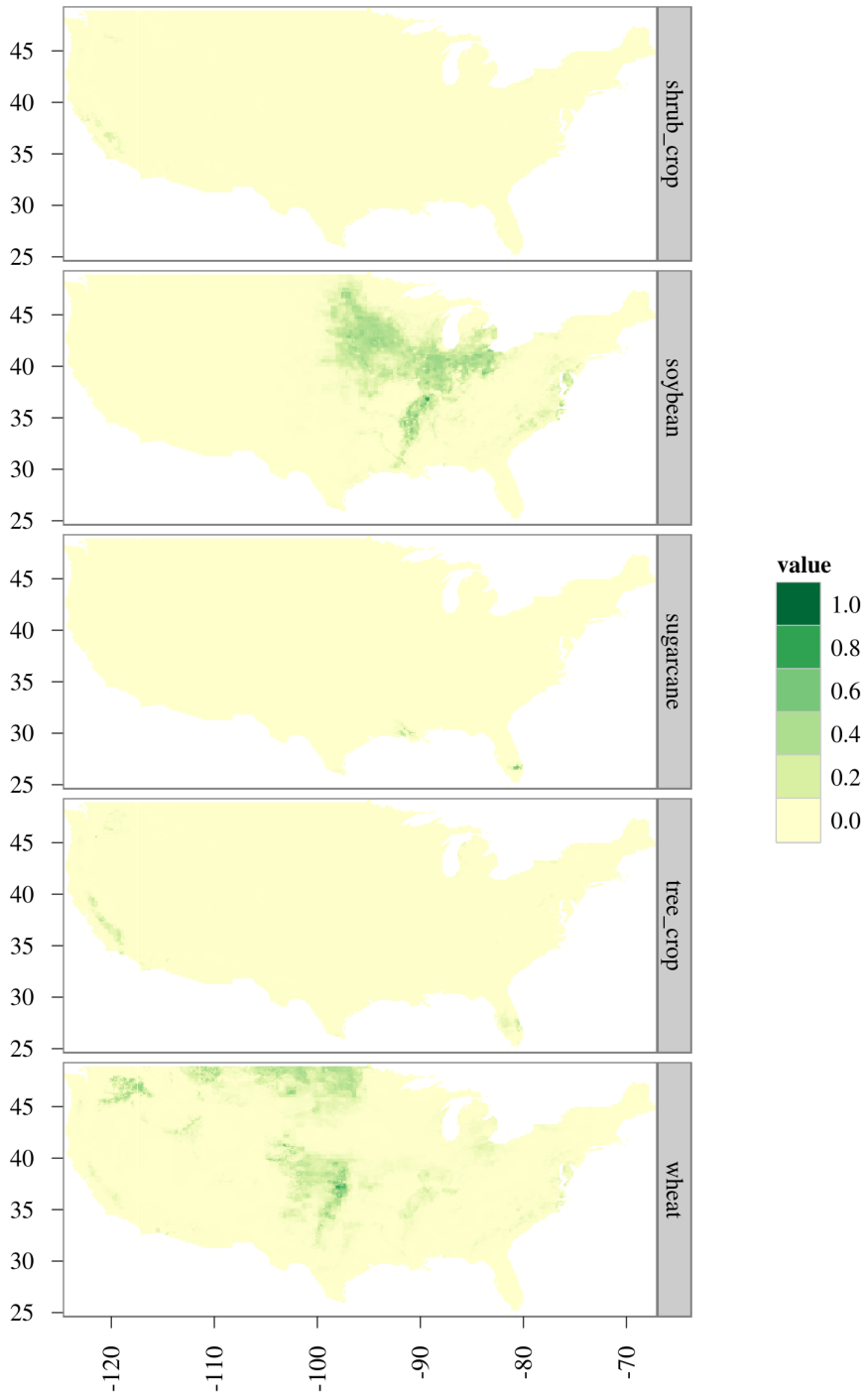

Figure 2.29: 175Crops2000 category maps (cont.)

Chapter 3

Analysis

In this chapter we will describe a procedure for combining information from the data sets described in chapter 2 using the same sub-pixel analysis data structure at 5′ resolution for the conterminous USA9 (cUSA) to produce a data set that exhibits high accuracy in the distribution of agricultural production according to the Agland2000 data set introduced in section 2.3, that provides a realistic characterization of other uses and covers as suggested by MLCT data from section 2.1 and particular aspects of the NLCD from section 2.2.

In this chapter we will evaluate the progress of the analysis in terms of areas given both in millions of acres (Ma) and millions of hectare (Mha). It is important to note that these areas cannot be computed directly from the geographic grid in which the data is contained and our maps are rendered. Because these 5′ grid cells are actually sections of a spheroid projected onto a plane, the areas that a given cell encompasses is not constant. A first-order approximation of these areas can be obtained as a function of the earth's mean radius and the cosine of a cell's latitude. In this way areas are maximum at the equator and approach zero as position approaches the poles. Conveniently Hijmans' `raster` package for the `R` analysis software provides a function `area()` that accepts any raster data set in geographical coordinates (longitude, latitude) as an input, producing a new raster data set whose values are the areas of the former in km^2. It is a simple matter to convert these to acres by subsequently scaling that result by a constant. See the source code in the appendix for further details.

We start by tabulating the aggregate areas by PEEL class for the data sets that we are using as inputs, MLCT, NLCD, and Agland2000. We evaluate the accuracy of cropland distribution in the MLCT data as a function of two values of the A_{min} parameter, which is defined as the minimum fraction of an MLCT pixel at its native resolution assigned to the primary class prior to aggregation to the 5′ PEEL model analysis grid. $A_{min} = 1.0$ represents consideration of only the primary class. $A_{min} = 0.5$ indicates that in the hypothetical situation of zero classification confidence the primary class would be assigned half of the area of that particular MLCT pixel, therefore this value represents maximum incorporation of the secondary class in our analytical framework, as modulated by the MLCT classification confidence data. These intermediate results are compared on the basis of root mean squared error (RMSE) metrics calculated relative to the distribution of cropland given in the Agland2000 data at the end of section 3.1.

In section 3.2 we describe a method for selectively incorporating cover fractions for particular classes from the NLCD data set due to a perceived underestimation of those classes by MLCT due primarily to its

lower resolution. Those classes are water, wetland, and urban. In the PEEL classification the "urban" class is broadened to include rural infrastructure that MLCT effectively counts as cropland. Accepting NLCD's quantification of these classes as truth is intended to counteract a perceived overestimation of cropland area in MLCT caused in part by a discrepancy in formulation of these data sets, Agland2000 representing actual harvested areas and MLCT catching up lots of ancillary land that may be associated with cultivated land but is not directly involved in crop production. The result is an adjusted version of the MLCT data as amended by these NLCD offsets.

Section 3.3 presents the results of fusing our adjusted MLCT map with the Agland2000 data by accepting the Agland2000 value for cropland as truth where possible and scaling other classes proportionally to describe the remainder of the landscape for purposes of the PEEL model. This operation is constrained by our decision to retain the NLCD offsets as firm figures for those classes in order to account for varying degrees of infrastructure development represented by the so-called urban class and water or wetland features not resolved in the MLCT data. Where Agland2000 conflicts with this constraint the cropland fraction is reduced accordingly.

Finally section 3.4 shows how information from the `175Crops2000` data set is used to disaggregate the cropland given by the result of the previous step in order to provide a rough characterization of the distribution of production of major crop commodities, corn (maize), soybean, wheat, rice, sugarcane, other cereals, and other field crops. This is important for the PEEL model because the intent is to model transitions in production in response to forecast commodity prices in addition to other drivers.

Through the offset and fusion steps we use decreasing RMSE figures to show that our complete characterization of the landscape is improving in accuracy with respect to `Agland2000`, the census-based distribution of productive cropland so that when other cover classes are scaled the distortions are minimized.

3.1 Comparison of Aggregate Areas

After decomposing the mosaic class MLCT indicates 495.4 Ma (200.5 Mha) of cropland for $A_{min} = 0.5$ and 488.1 Ma (197.5 Mha) for $A_{min} = 1.0$ in the cUSA in 2001. Aglands2000 indicates roughly 446.5 Ma (180.7 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 75.4 Ma (30.5 Mha) of

	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
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
barren	0.0	24.5	32.8	28.9	32.8	28.9
urban	0.0	102.8	27.3	29.8	27.3	29.8
wetland	0.0	95.0	26.0	11.0	26.0	11.0
water	0.0	96.5	74.3	75.0	74.3	75.0
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

development remaining after subtracting the MLCT urban class from all developed classes in the NLCD after they have both been aggregated to the $5'$ 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 in chapter 2 was 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.

Figure 3.2 and Figure 3.3 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.

To examine the relationships between the distributions of cropland that we derive from the MLCT data relative to the Agland2000 data we will use "hexbin" plots which are essentially two-dimensional histograms that show the number of grid cells that occur within discrete regions of the space defined by coordinates that are cropland fractions for the two data sets. This operates much like a common scatter plot but for

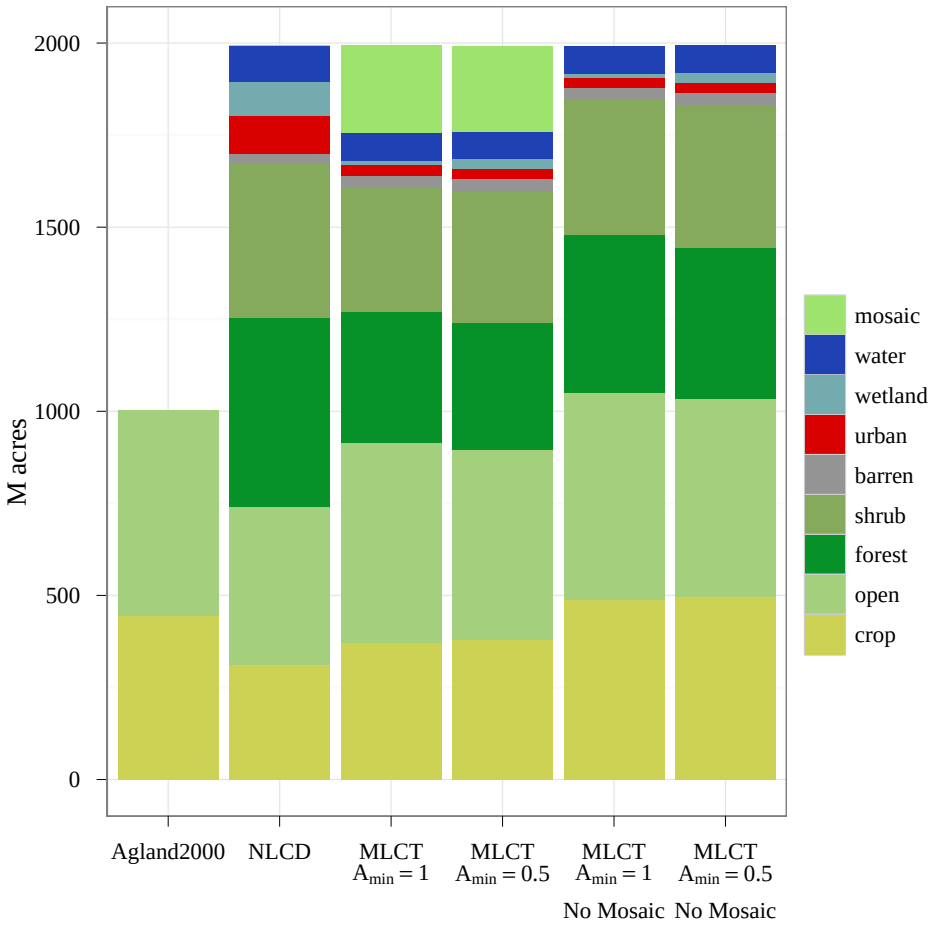

Figure 3.1: Total Acreages by Map and Cover

data sets with as many observations as we wish to include it gives a cleaner representation of that structure. For our plots we have chosen to employ a logarithmic scale because of the wide range of counts calculated for the bins. This gives a more complete picture of the overall dispersion and local concentration of the observations. Our first example of such a plot is Figure 3.4 which plots the crop fractions of MLCT with $A_{min} = 1$ versus those of the Agland2000 crop map. As one would expect there is an overall correlation among these variables, especially given that Agland2000 provides prior probabilities to the MLCT classification. It is clear that the MLCT primary class exhibits a positive bias overall, although a subset that is negatively biased is also apparent for low values of the Agland2000 crop fraction in the interval $[0.1, 0.5]$. Also of particular note is the drastic decrease in correlation when Aglands2000 reaches 1.0 relative to the stronger relationship over the interval $[0.8, 1.0)$. It is difficult to speculate on the nature of this structure, but suffice it to say that there is something peculiar about the Agland2000 allocation procedure

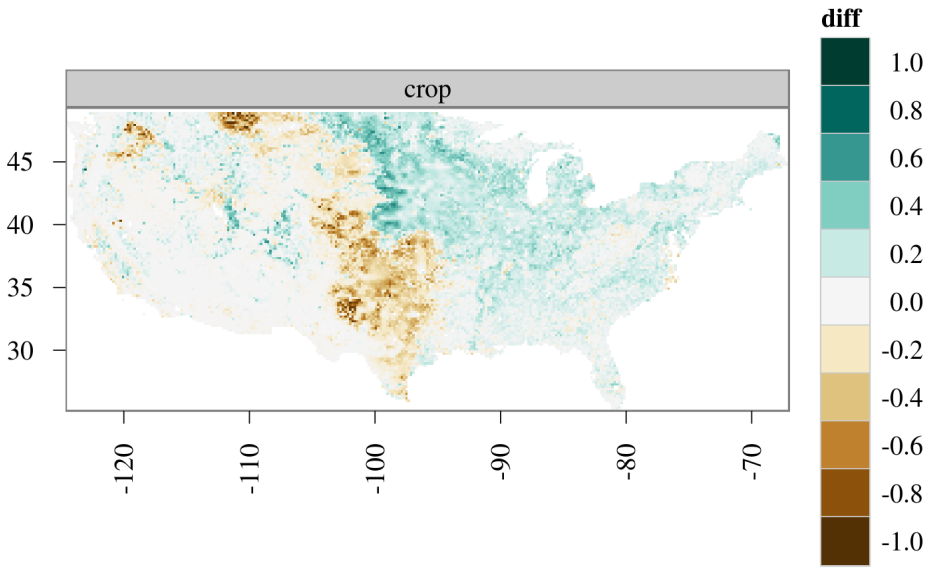

Figure 3.2: Difference between MLCT (no mosaic, $A_{min} = 1.0$) and Agland2000 crop

that drives the crop fraction to its maximum in areas where the remote sensing data clearly resists such a characterization. This may be caused by systematic errors in the agricultural census data that drive the Agland2000 algorithm forcing unrealistically high concentrations in order to satisfy the algorithm's constraints.

A_{min}	RMSE
0.5	0.165
1.0	0.180

Table 3.2: RMSE, MLCT vs. Agland2000 crop

We expect that setting $A_{min} = 1$ will produce a maximum overall bias and attendant error by assigning entire pixels to the cropland class and not allowing for the possibility of mixed covers. The results on Table 3.2 indicate that $A_{min} = 0.5$ is more representative of the distribution of cropland because although

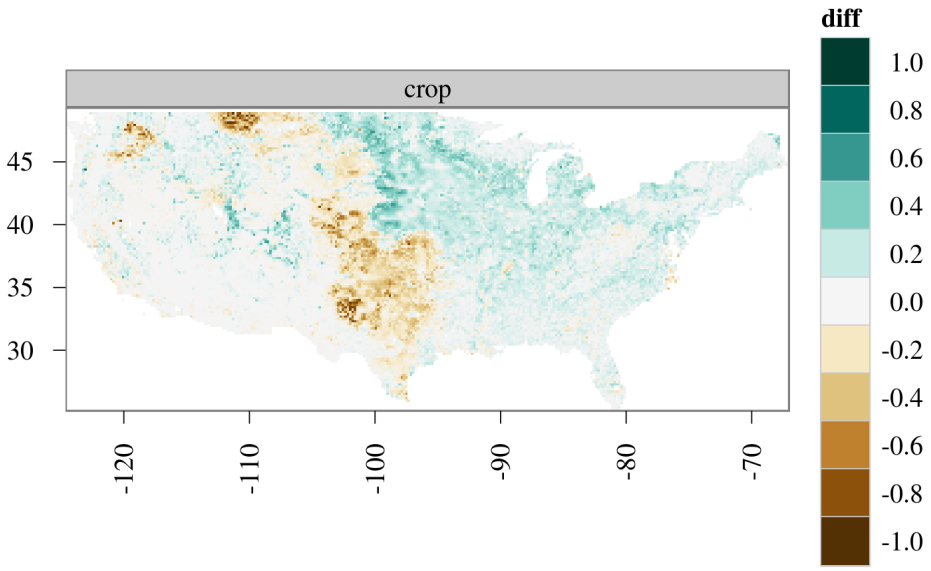

Figure 3.3: Difference between MLCT (no mosaic, $A_{min} = 0.5$) and Agland2000 crop

the total area indicated is higher according to Table 3.1, there is less error on a cell-by-cell basis indicating that it does a better job of representing the spatial distribution than $A_{min} = 1.0$. This is reflected in the structure revealed by Figure 3.5 where fewer cells in the MLCT data are set at 100% crop because of including the secondary class in calculating $5'$ coverage fractions. Where crop was included in a secondary class it also caused cells of near-zero value for MLCT to lift away from the x-axis. The uncorrelated observations for Agland2000 equal to 1.0 are still present, however. This result is adequate for our purposes to determine that our logic in considering the secondary class in the manner we have for $A_{min} = 0.5$ is correct. 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 to this improved fit with Agland2000 cropland and its full consideration of all information imparted by the MLCT data.

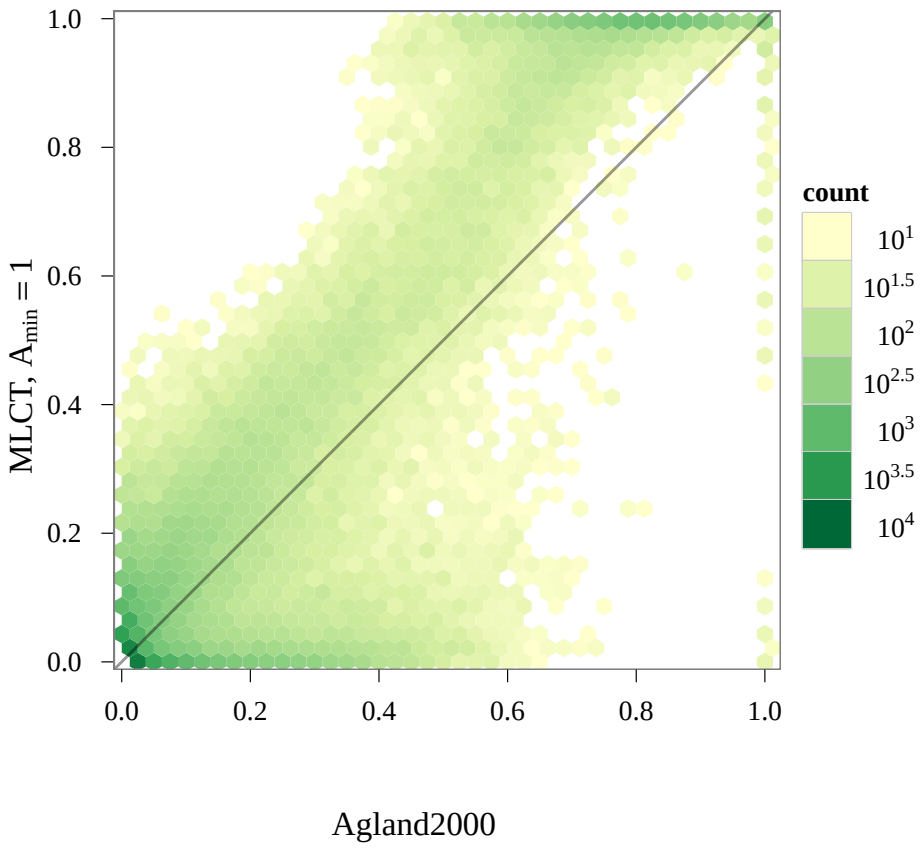

Figure 3.4: Hexbin plot of MLCT crop ($A_{min} = 1.0$, no mosaic) versus Agland2000 cropland

3.2 NLCD Offsets

From Table 3.1 it is apparent that the MLCT results are negatively biased in the total 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 MLCT data due to its resolution. The most obvious example is the rural transportation networks in areas surveyed under the Public Land Survey System (PLSS) where roads have been laid out on a generally regular grid of square miles. In the PEEL classification this infrastructure is included in the urban class as another form of developed land, perhaps making "urban" somewhat of a misnomer, but it hails to its origins in the IGBP classification scheme and provides a short label, a great convenience in programming. It is important to represent wetlands and water features in our input to the PEEL model because these areas have high likelihoods of being set aside for conservation purposes, which would be represented as a con-

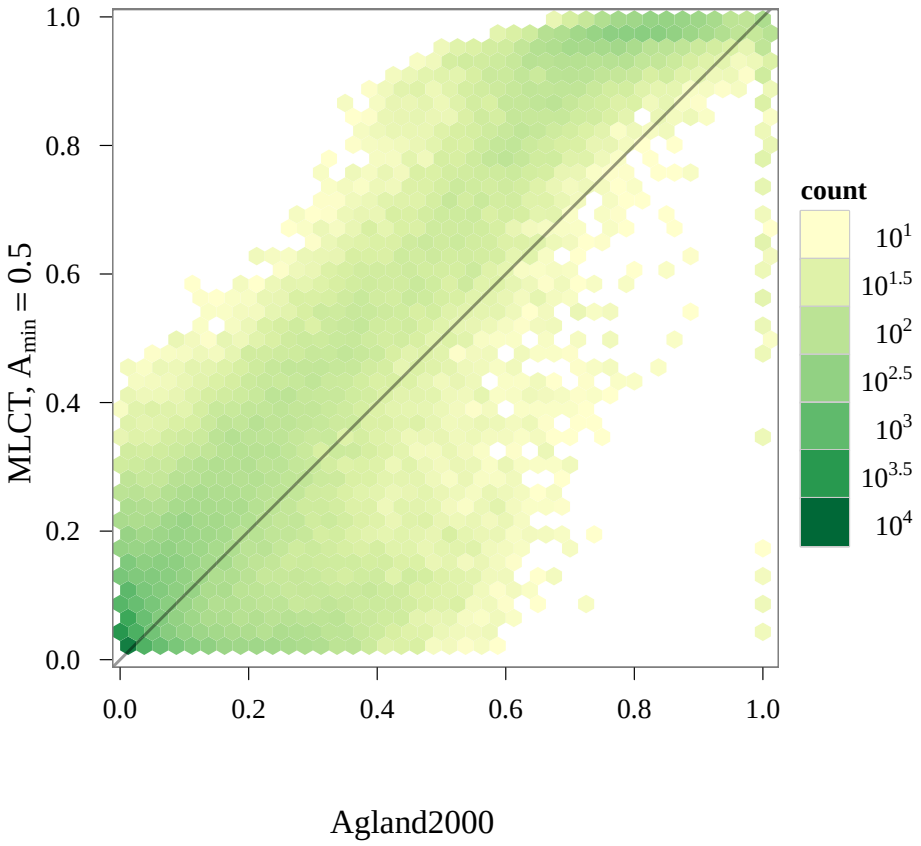

Figure 3.5: Hexbin plot of MLCT crop ($A_{min} = 0.5$, no mosaic) versus Agland2000 cropland

straint on land conversion in the model. In the event that NLCD overestimates these areas it would be an acceptable error to carry over to the PEEL model in order to be conservative in allowing for conservation measures in a greater number of grid cells, absent more precise LULC data with respect to the water and wetland classes.

To merge this information from the NLCD we begin by simply accepting the areas for water, wetland, and urban classes in the reclassified, 5′-aggregated version of NLCD that we have computed as truth and calculate offsets for those classes versus our 5′ MLCT data by straight subtraction. Where NLCD is greater the difference will be positive and so a positive offset will be added to the fraction already present for any one of the "truth" classes from NLCD. The other classes are then adjusted so that they are present in proportion to each other as indicated by MLCT but in the area remaining after accepting the water, wetland, and urban areas from NLCD. The additive offsets needed to achieve this balance and account for the en-

tire area of the cell are calculated so that the effects of this process on all classes may be considered on a common basis.

For the calculation of the offsets we drop back to the result of aggregating the MLCT data to $5'$ with $A_{min} = 0.5$ prior to mosaic decomposition. Presumably there are rural roads comprised of 30m NLCD pixels cutting through the lower-resolution MLCT pixels including those classified as mosaic. In fact, by its very nature as a hybrid class made up of natural cover and agricultural land use we expect roads to be an important component of the landscape. Figure 3.6 and Figure 3.7 show the spatial distributions of the offsets calculated based on our assumptions about the water, wetland, and urban classes in the NLCD. We have verified that these offsets sum to zero for each grid cell. Any area deducted from one class must be added to one or more classes in the same cell in order to conserve the total area and maintain the sum of the fractions at 1.0.

The maps of these offsets are shown using a logarithmic scale in order to bring attention both to areas of significant adjustment, greater than 10%, as well as to show the extent to which small adjustments on the order of 1–5% occur. From these maps we can see the detailed structure of drainage networks in the water class and population centers in the urban class which could easily be confused with the vegetative classes in the MLCT classification. This refers, for example, to heavily wooded suburbs where transportation infrastructure is obscured and difficult to resolve. The offsets for the NLCD truth classes are generally positive, although not strictly so because the algorithm does not preclude the possibility that MLCT may locally overestimate these classes in particular regions and still suffer an aggregate deficit relative to NLCD.

Figure 3.8 shows the result of calculating a matrix of correlations among the offsets calculated for each class based on NLCD. Each cell in the matrix reflects the value of the statistical correlation between the corresponding classes within the NLCD offset maps resulting from the algorithm described above. This gives us an overall sense of the effect of applying these offsets by showing which changes are strongly correlated, whether it be positively or negatively. It is a symmetric matrix because the classes on both axes are from the same data set and any single classes is, of course, perfectly correlated with itself. Going in we would expect to see negative correlations between classes accepted as truth from NLCD, water, wetland, and urban, and the other classes because the purpose of applying these offsets was to bring the total areas of these classes up, which can only happen at the expense of the other classes. For example the wetland offsets show strong negative correlations with forest, shrub, and mosaic. This stands to reason as a likely problem with classification due to fundamental differences in the remote sensing data such as resolution, the interpretation thereof, or disagreement/overlap in class definitions. Many areas of forest and shrub

land can exhibit properties of a wetland when standing water and high soil moisture are persistent. The "NLCD truth" classes' offsets are positively correlated with one another because they are generally positive everywhere. Likewise, non-truth classes are positively correlated with one another because they are all being assigned negative offsets to make room for the increased values of water, wetland, and urban fractions. The crop and mosaic classes are most strongly negatively correlated with urban which reflects the widespread adjustments to account for rural transportation networks and smaller settlements.

The resulting offsets are added to the aggregated fractions calculated from the MLCT with $A_{min} = 0.5$. The mosaic decomposition step is readily applied to the adjusted data set because the adjusted fractions are fundamentally in same form as the intermediate form of the MLCT data calculated in subsection 2.1.3, only the values have changed.

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 3.3.

offset	A_{min}	RMSE
TRUE	0.5	0.151
FALSE	0.5	0.165
FALSE	1.0	0.180

Table 3.3: RMSE, MLCT vs. Agland2000 crop with NLCD offsets

Seeing that this modification 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.

	Agland2000	NLCD	MLCT	MLCT No Mosaic	NLCD Offsets	MLCT Adjusted	MLCT Adjusted No Mosaic
water	0.0	96.5	74.3	74.3	22.3	96.5	96.5
forest	0.0	513.2	344.7	410.8	-44.7	300.1	355.7
shrub	0.0	420.1	358.7	387.2	-23.8	334.9	358.0
open	557.1	429.6	516.9	538.7	-21.0	495.9	514.9
wetland	0.0	95.0	26.0	26.0	69.0	95.0	95.0
crop	446.5	310.8	378.9	495.4	-39.0	339.9	437.6
urban	0.0	102.8	27.3	27.3	75.4	102.8	102.8
mosaic	0.0	0.0	232.9	0.0	-37.4	195.5	0.0
barren	0.0	24.5	32.8	32.8	-0.9	31.9	31.9
(all)	1003.7	1992.5	1992.5	1992.5	-0.0	1992.5	1992.5

Table 3.4: Effect of NLCD offsets on total acreages, $A_{min} = 0.5$

Figure 3.9 shows the totals by class of the offsets that result from this calculation. The item labeled "total" appears blank because a value of zero is plotted there indicating that area was conserved in this op-

eration, which is to sat that area subtracted from one class was reallocated to another. As expected, the most significant offset was for the urban class, representing the low-density infrastructure outside of concentrations of development large enough and dense enough to be identified in the MLCT classification. Water and wetland fractions were also increased to bring the total areas of those classes in line with NLCD. However, the most important outcome with respect to our stated objective of bringing total cropland areas in line with the total from Aglands2000 is the reduction of crop areas by 39 Ma (15.8 Mha) and mosaic areas by 37.4 Ma (15.2 Mha). This will result in a total reduction of 57.8 Ma (23.4 Mha) of the final crop class after mosaic decomposition because mosaic land is taken to be half cropland by definition. Figure 3.10 shows the effect of adding these offsets and subsequently performing the mosaic decomposition operation, which brings the cropland area for the PEEL input data set into closer agreement with Aglands2000, 437.6 Ma (177.1 Mha) and 446.5 Ma (180.7 Mha) respectively. The significance of this result is that it is in no way conditioned by the desired cropland area estimate, rather shows a convergence in these estimates by selectively incorporating information about other classes from an independent data set, namely NLCD.

3.3 Fusion of Adjusted MLCT and Agland2000

By bringing the aggregate crop areas of our MLCT-derived data set into greater agreement with Agland2000 and accepting the water, wetland, and urban fractions indicated by the NLCD we are now ready for the final manipulation of the remaining classes, forest, shrub, open, and barren, that will bring our complete data set into maximum agreement with Agland2000. To do this we will be setting the crop fraction equal to Agland2000's crop value everywhere that this does not conflict with the allocation indicated by the NLCD offsets. Anywhere that there is a conflict such that Agland2000 indicated more cropland than allowed by the NLCD offsets the crop fraction will be set to one minus the total of the offsets and all other classes will be set to zero. Otherwise the non-offset, non-crop classes will be scaled to fit proportionally into the remaining area left after incorporating those classes which we have given primacy. The assumption in this step is that the census data behind the Agland2000 crop map is a "ground truth" and Ramankutty's method for allocating that area within the $5'$ grid is sufficiently faithful to that truth. The one difficulty of note in this step is the presence of cells where we have MLCT/NLCD-derived fractions for LULC but Agland2000 is null, that is to say that no data is given indicating that those cells were not included in the land mass. In those cases we accept the crop fraction previously calculated since there is nothing to compare it against.

agland	offset	A_{min}	RMSE
TRUE	TRUE	0.5	0.017
FALSE	TRUE	0.5	0.151
FALSE	FALSE	0.5	0.165
FALSE	FALSE	1.0	0.180

Table 3.5: RMSE of PEEL vs. Agland2000

	Agland2000	NLCD	MLCT No Mosaic	MLCT Adjusted No Mosaic	PEEL
water	0.0	96.5	74.3	96.5	96.6
forest	0.0	513.2	410.8	355.7	380.9
shrub	0.0	420.1	387.2	358.0	362.8
open	557.1	429.6	538.7	514.9	479.1
wetland	0.0	95.0	26.0	95.0	95.0
crop	446.5	310.8	495.4	437.6	443.7
urban	0.0	102.8	27.3	102.8	102.8
mosaic	0.0	0.0	0.0	0.0	0.0
barren	0.0	24.5	32.8	31.9	31.7
(all)	1003.7	1992.5	1992.5	1992.5	1992.5

Table 3.6: PEEL acreages, $A_{min} = 0.5$

Figure 3.15 shows a thematic map that classifies the cells in our study area according to their agreement on the cropland fraction between Ramankutty's Agland2000 data set and the newly created PEEL data set. The first class indicated by "PEEL = 0" in the legend represents where Agland2000 cropland fraction is zero so there is no potential for conflict. The second class shows where the PEEL crop fraction is greater than zero but Agland2000 is null, meaning no data was given for those cells. Such cells generally occur in coastal areas and on the shores of the Great Lakes, reflecting that Rmankutty's criteria for counting a cell as "dry land" was somehow more restrictive. The third class shows where the NLCD "truth" classes (water, wetland, urban) allowed us to bring the PEEL crop fraction in line with Agland2000 without violating the assumption that those fractions should be carried over from the NLCD aggregation. The fourth class reveals where those constraints could not be simultaneously satisfied. Those cells correspond to the bins in Figure 3.14 that fall below the equality line because the values from the NLCD offsets are given precedence and the crop fraction is limited accordingly, which of course might mean that other non-offset, non-crop classes could be summarily reduced to zero. The final class highlights pixels that Agland2000 assigns a crop fraction of 1.0 which seems unrealistic given that some infrastructure and uncultivated cover must be present within such large areas.

3.4 Disaggregation of PEEL Crop Fractions According to 175Crops2000

We could assume that forage crops come from open class but we don't know enough about the confusion between Aglands2000 pasture and the open class in the first place, much less to make an informed speculation about how forage crops would be classified by MLCT. The focus here is field crops so that is the only class that we are attempting to disaggregate and forage crops are included there for now. Tree and shrub crops could be taken from the corresponding cover types, but assuming that they are caught up in that classification is a blind leap and their areas are small. On the other hand, their economic impact may be disproportionate to their areas by virtue of price, but this will have to be studied more carefully. Double-cropping is ignored for now by normalizing the crop fractions by the sum of all crops, which can exceed unity in instances of intense double-cropping. The predominant double-cropping system in the cUSA to our knowledge is soy followed by winter wheat, but there may be others such as multiple cropping of rice in the southern extremes of its range. In areas where soy and wheat are double-cropped their areas will be underestimated in this data set relative to that given in the 175Crops2000 data set, subsequent to the NLCD offset adjustment. This issues also bears further study.

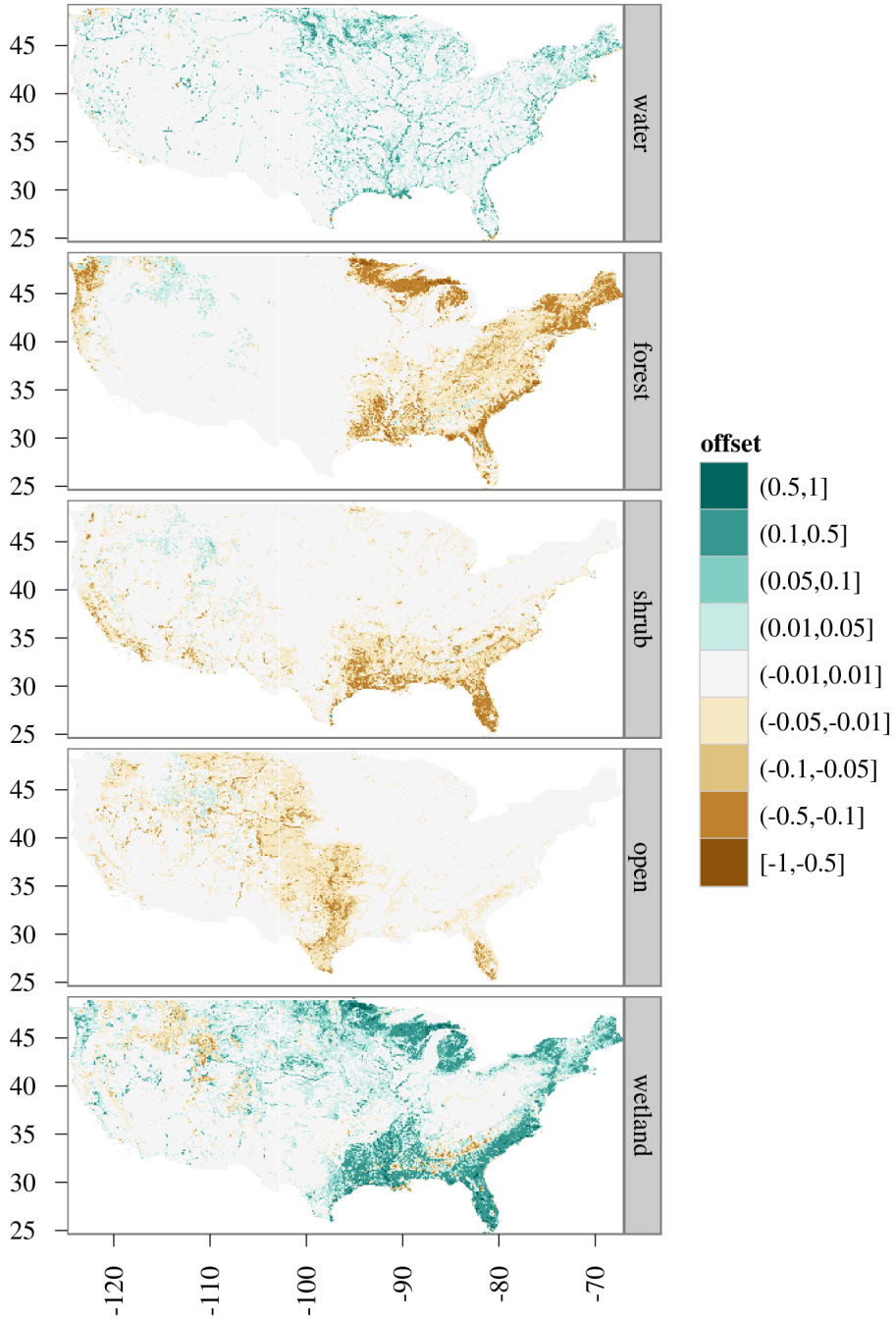

Figure 3.6: NLCD offsets

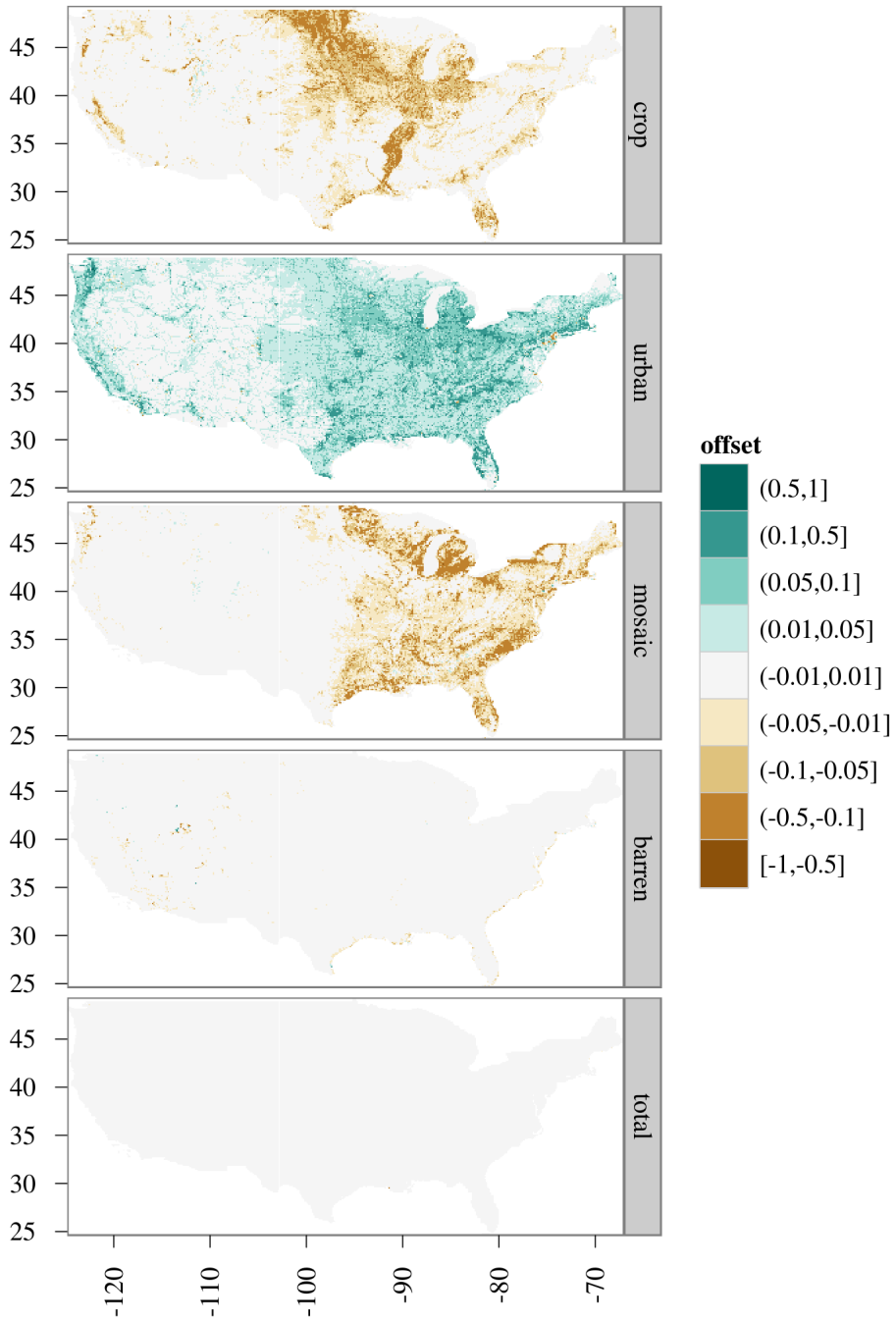

Figure 3.7: NLCD offsets (cont.)

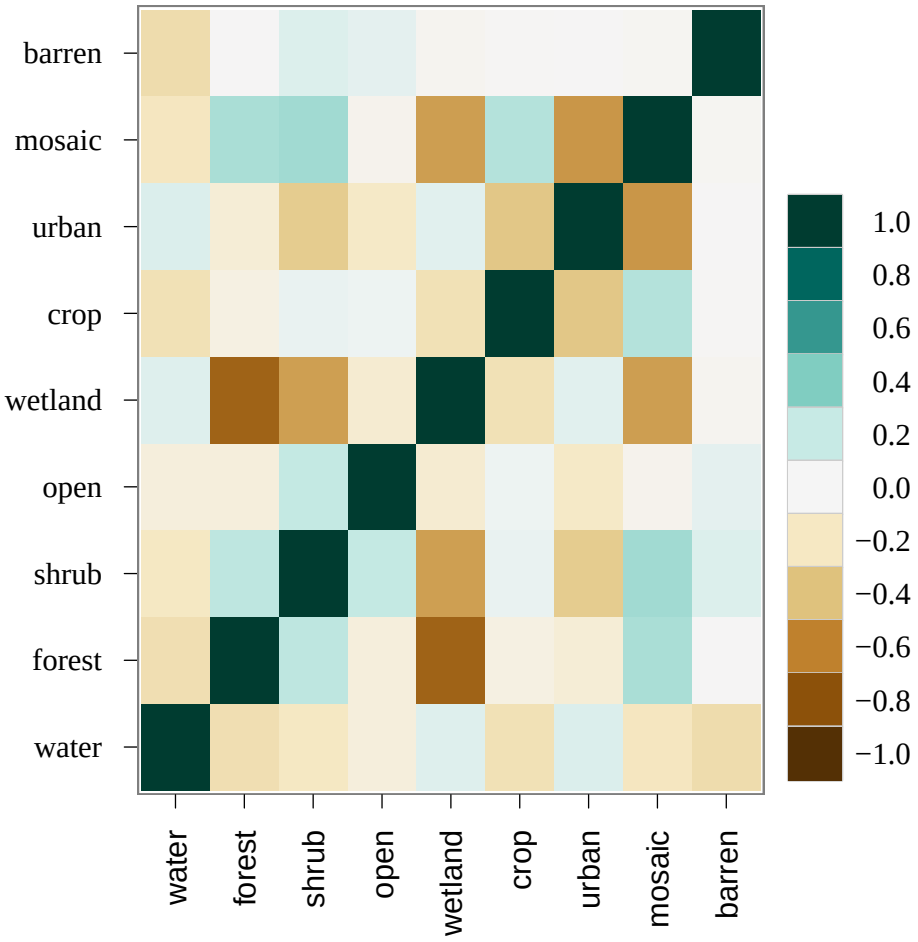

Figure 3.8: Correlation matrix of NLCD offsets

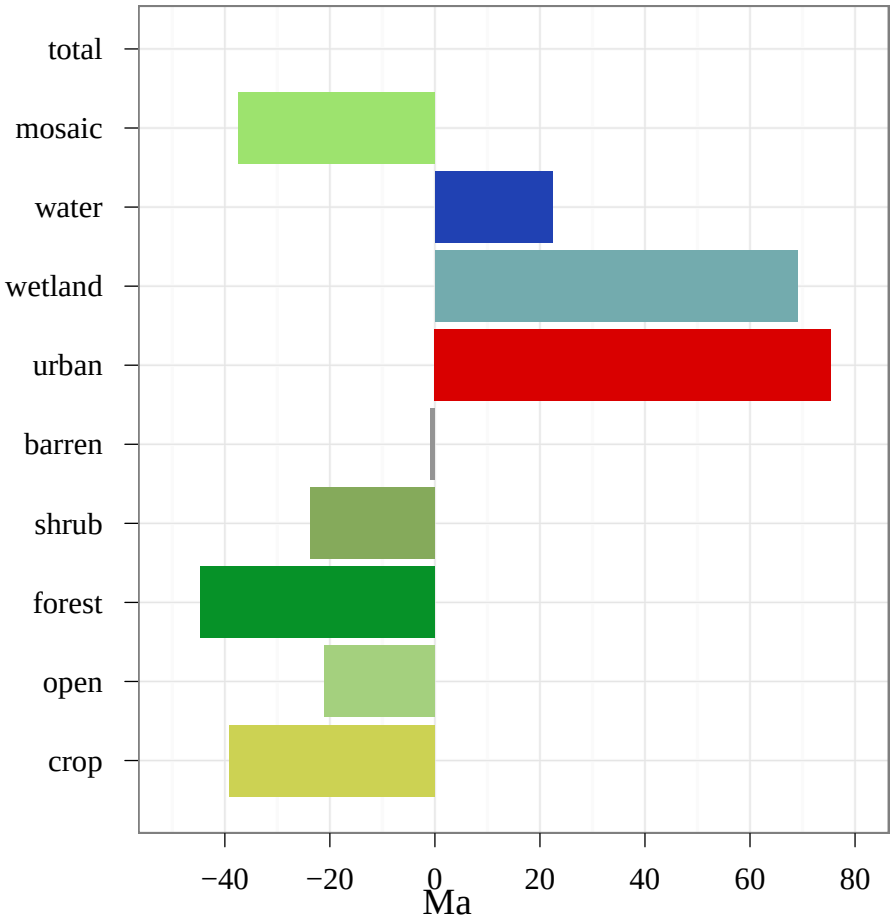

Figure 3.9: Total offsets calculated from NLCD

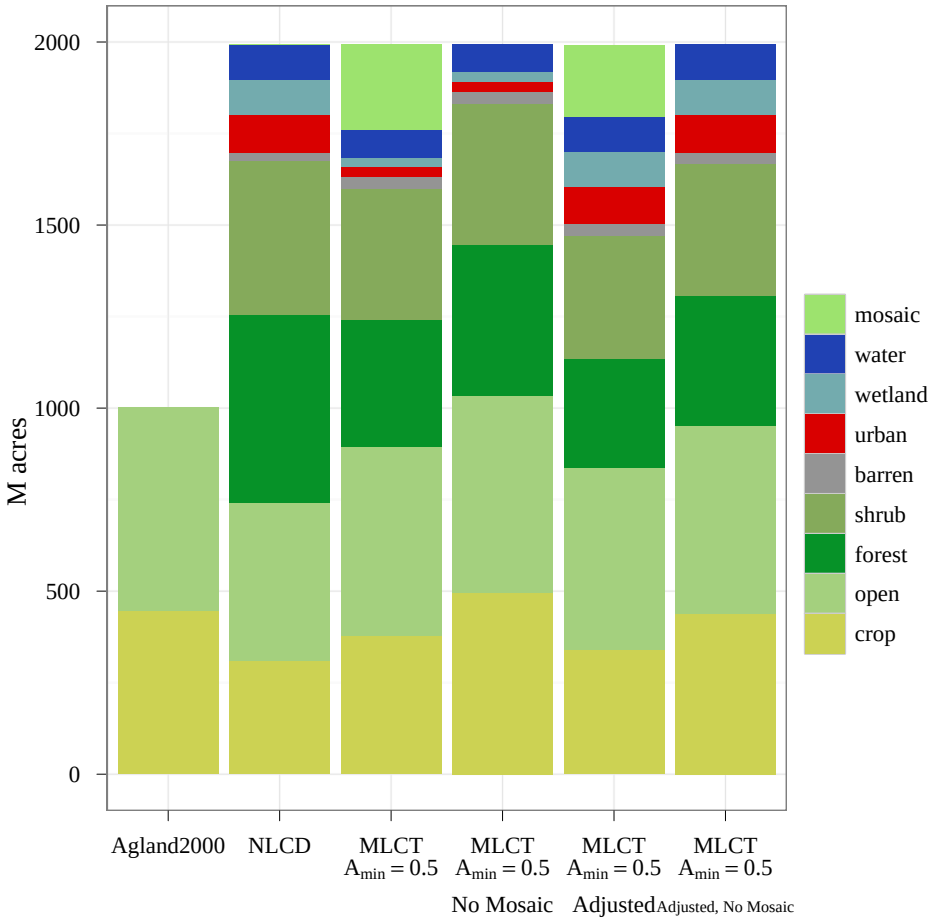

Figure 3.10: Total acreages after NLCD adjustment

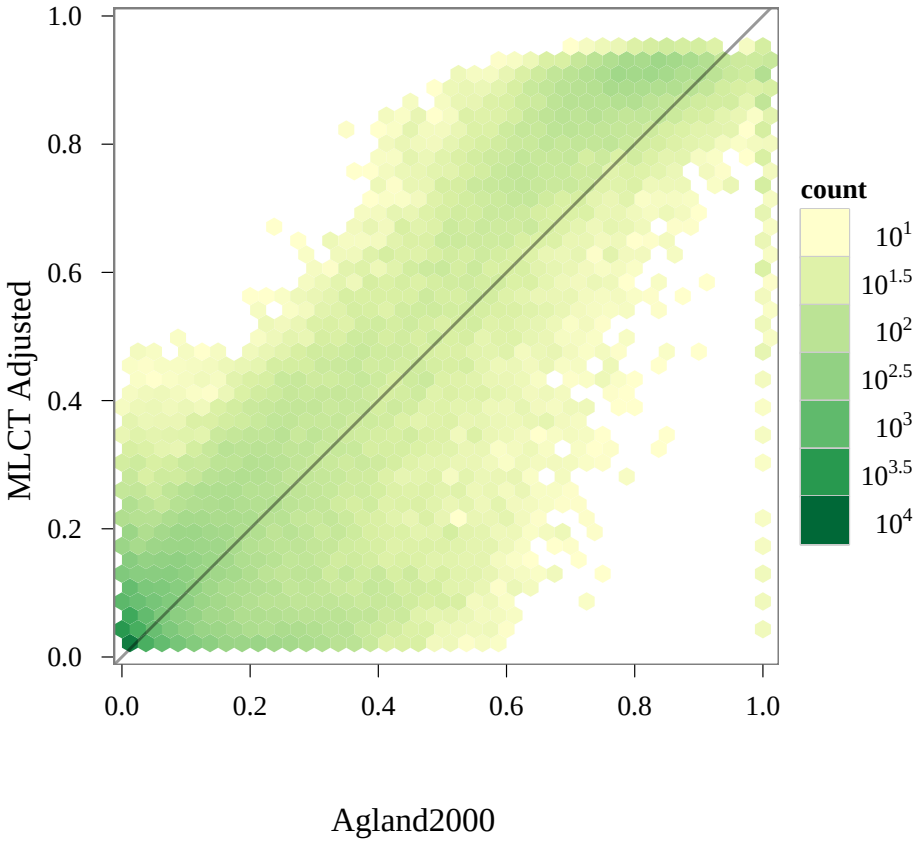

Figure 3.11: Hexbin plot of MLCT adjusted crop versus Agland2000 cropland

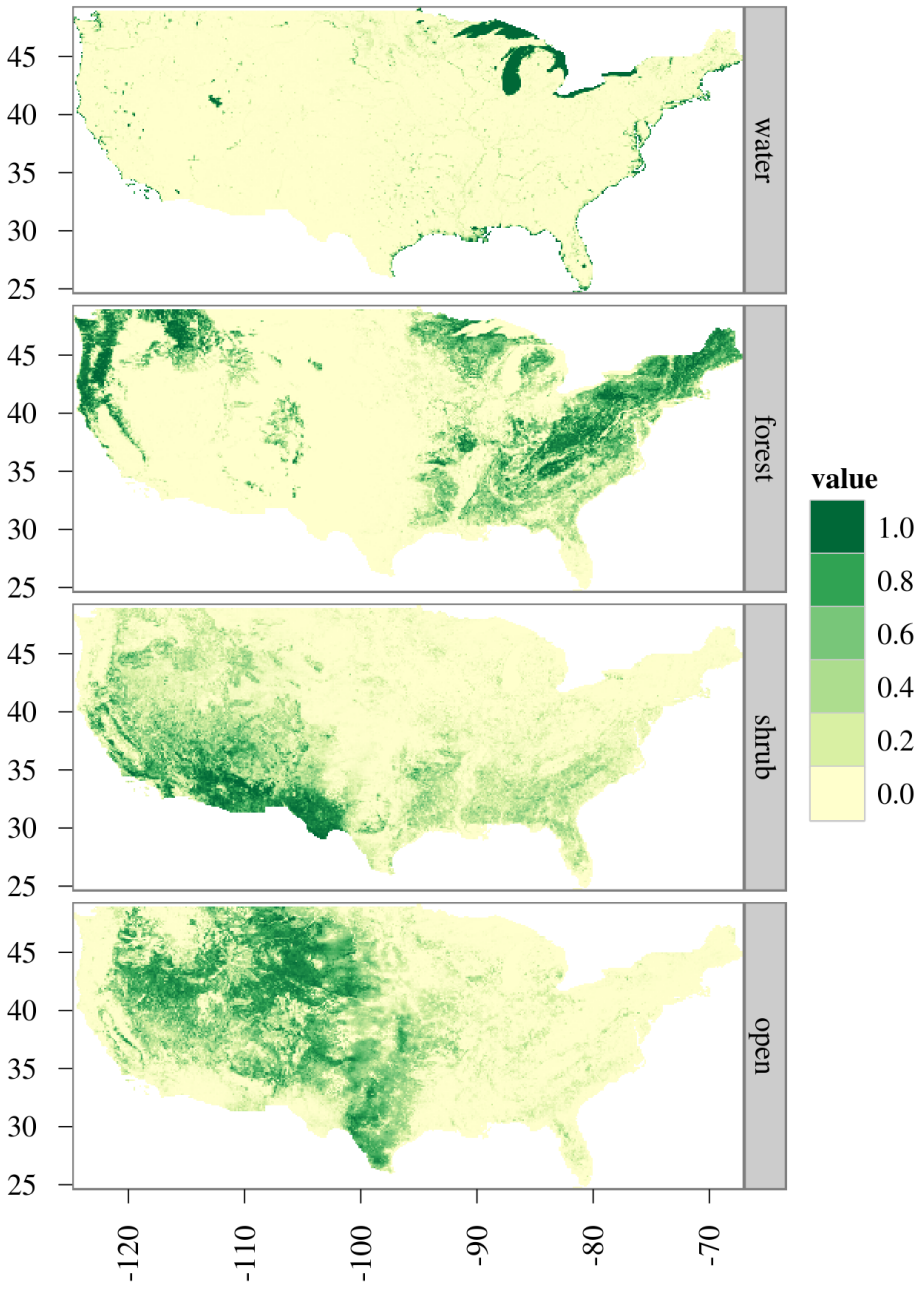

Figure 3.12: Final PEEL maps

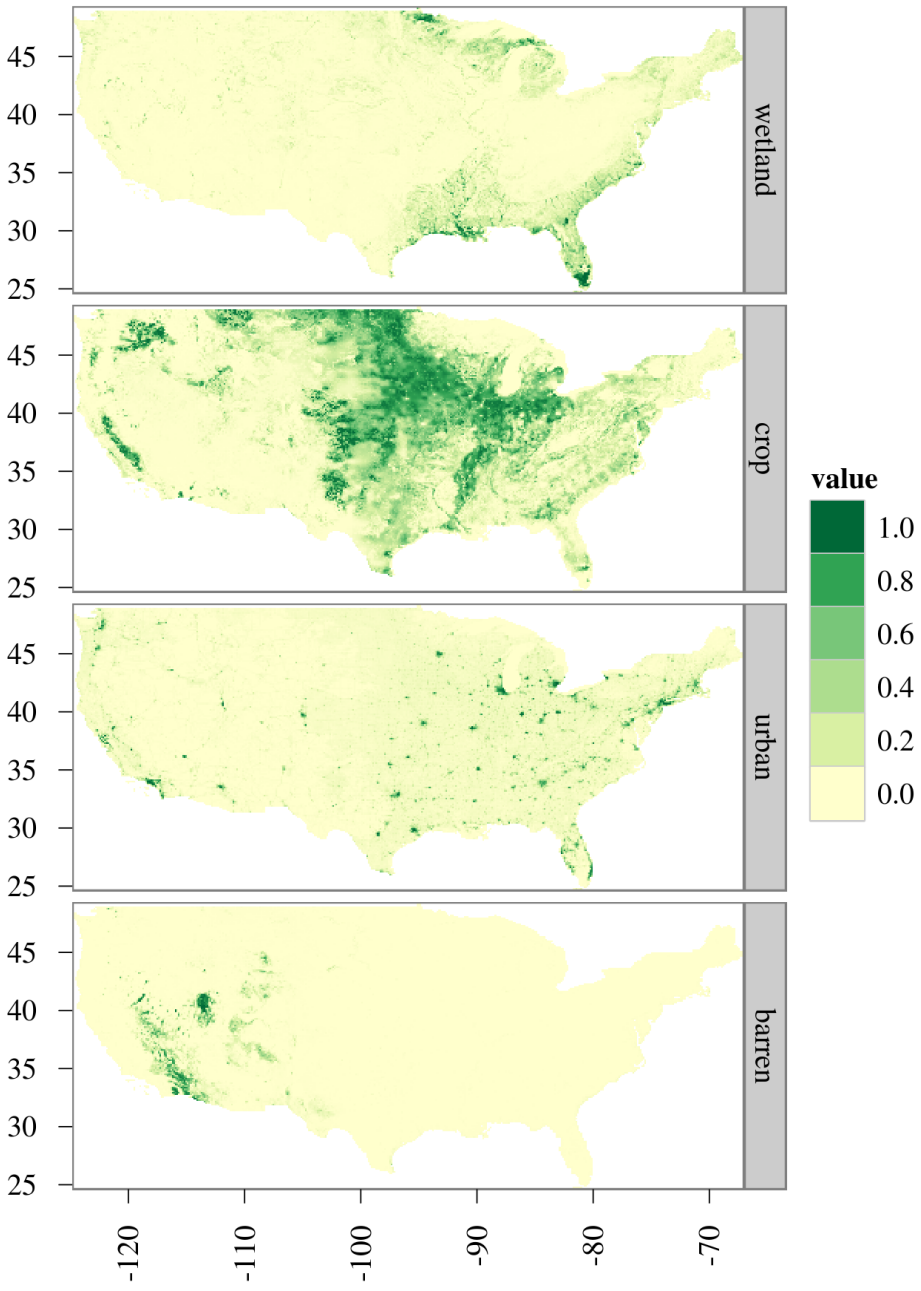

Figure 3.13: Final PEEL cover maps (cont.)

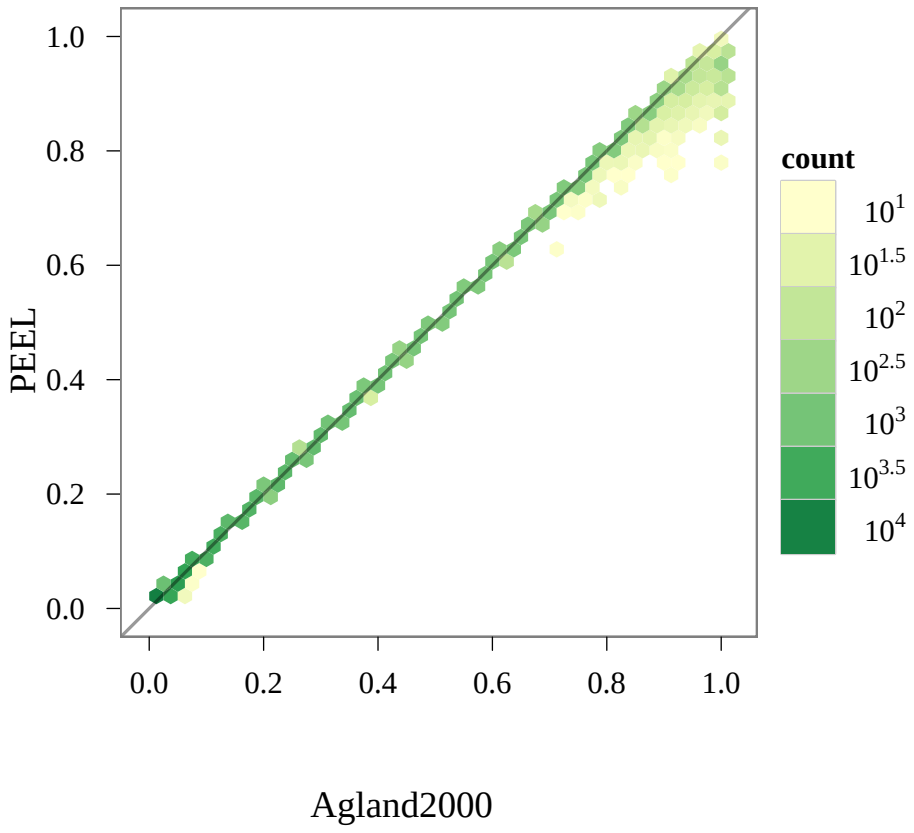

Figure 3.14: Hexbin plot of PEEL crop versus Agland2000 crop

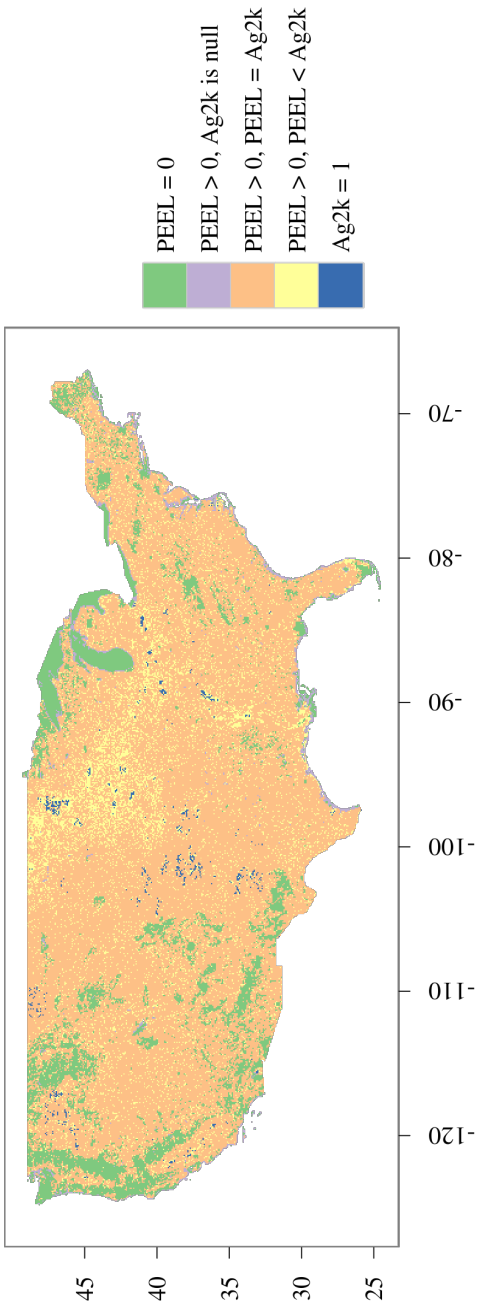

Figure 3.15: Conflicts between NLCD offsets and Agland2000

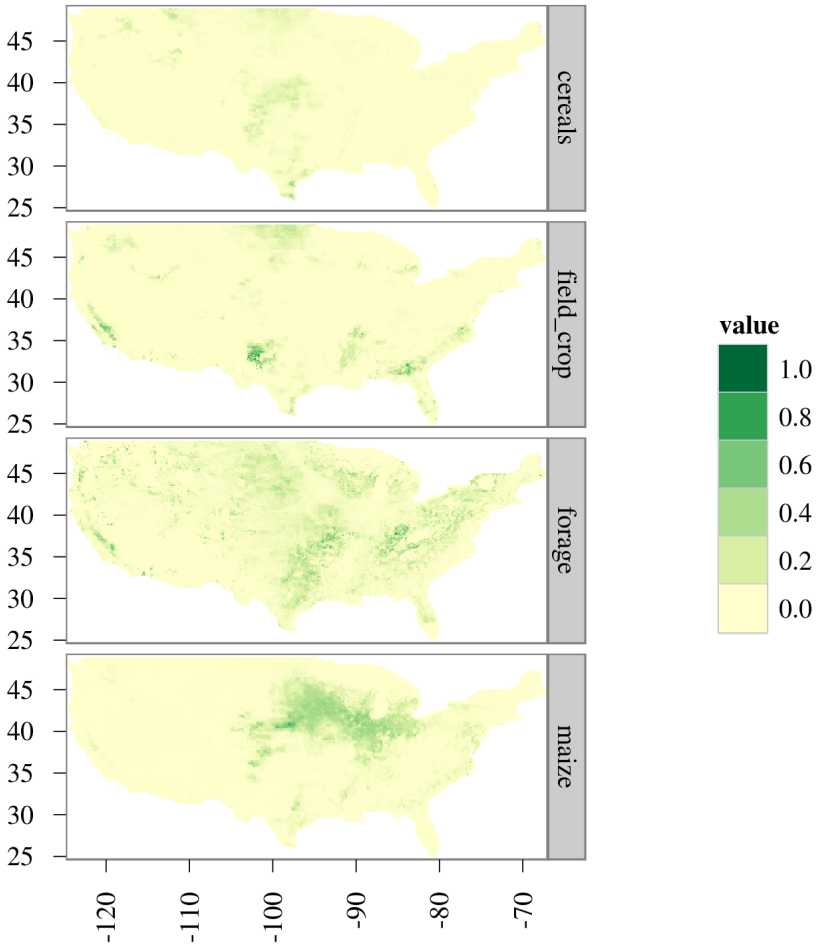

Figure 3.16: Normalized fractions for crop sub-classes

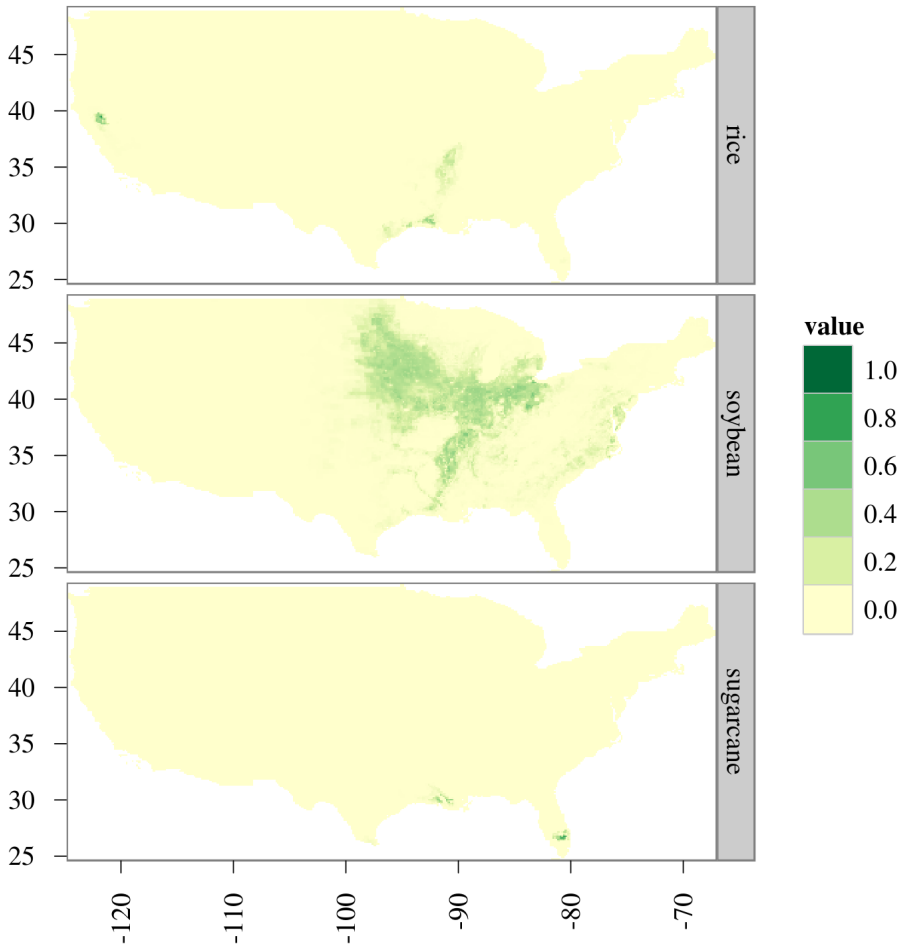

Figure 3.17: Normalized fractions for crop sub-classes (cont.)

Chapter 4

Conclusions

The goal of this study was to produce a LULC data set that was as accurate as possible with respect to its representation of the distribution of agricultural production and that also offers a reasonable characterization of non-crop covers and land use beyond those agricultural uses. We have accomplished that. In doing so we have adopted a sub-pixel data structure for conveying land use and land cover information, albeit at a low spatial resolution by today's remote sensing and GIS standards. However, we hope that our readers will consider that this data structure has particular mathematical properties that may make it useful for their applications. The ability to perform raster algebra on these stacks of maps made it possible to apply scaling factors and offsets with concise syntax in the `R` language that would not have been feasible in a discrete, categorical framework.

We maintain that the reproducible research aspect of this study was critical to its success. By elaborating the analysis in an interactive environment where every component of every data structure is subject to inspection many missteps were discovered in the course of our work. In other GIS analysis environments it might have been too difficult to perform basic sanity checks on intermediate outputs before moving on and too easy to rely on a sense of "everything looks right" in a GUI environment. The ability to point to a body of source code that expresses the steps of the analysis is a poignant example of reproducibility, a pillar of the scientific method that has fallen out of vogue because of the complexity of modern spatial analysis until the recent emergence of applicable software tools coupled with adequate computing resources.

This analysis would not have been possible without the `raster` package for `R` developed by Hijmans, van Etten and other contributors. We consider this interface to geospatial raster data sets in the `R` statistical analysis environment an important contribution to spatial analysis and a laudable accomplishment because it unleashes the power of a sophisticated, popular, open-source, free software programming language for statistical operations on large geospatial raster data sets. We expect that this demonstration will foster additional interest in and use of this software, as well as contribution to its continuing development. Directions for possible enhancement that would increase the utility of this package based on our experience include streamlining a truly functional programming interface by improving on the existing `overlay()` function, harnessing available `R` extensions for parallelism and porting core functionality to a `C` or `C++` library to improve performance, and improving the visualization interface, perhaps through application of the `ggplot2` package as we have done here.

In the CIM-EARTH/PEEL research agenda the next logical extension of the envisioned land use transi-

tion model would be to apply it to a global study area. Once a method for creating a global initialization data set for the year 2001 is formulated we would like to apply that method to the subsequent years of the MLCT time series. In order to do so we would need ancillary data that spatially extends the aspects of the NLCD that we have employed and temporally extends the information given in the Ramankutty & Monfreda data sets. At this time a method for calculating the correction offsets for over-estimation of cropland and under-estimation of inland water features, wetlands, and development/transportation infrastructure over a wider area with greater time depth has not been identified. Only with this information in hand were we able bring MLCT cropland and Agland2000 cropland into close enough agreement to minimize the mathematical manipulation necessary to reasonably quantify the non-crop cover and use classes with the desired fidelity to the best-available rasterized agricultural census data.

In the absence of high-resolution data on rural development, being the low-density portion of PEEL's "urban" class that falls below MLCT's detection threshold we propose that it might be possible to model the over-estimation factor of the MLCT cropland class. This factor would be defined as the ratio of total area encompassed by the MLCT cropland classification to acreage actually under cultivation and could potentially be modeled as a function of classification confidence and secondary class using the data described and produced here as a training set. The null hypothesis in the formulation of such a model is that there is enough diversity among agricultural landscapes in our cUSA study area to adequately characterize agricultural landscapes around the world in this regard. Similarly it might be possible to directly model the "urban" percentage below the MLCT detection threshold as a function of population density and agricultural productivity, identifying said threshold in the process. There is a clear dependency between these offsets in agriculturally productive regions so modeling them in conjunction somehow may be constructive. We expect that global offsets for the water and wetland classes will be harder to obtain without corresponding proxy statistics with which to formulate a model but perhaps we can expect greater availability of spatially explicit catalogs of ecological services and sensitive/protected areas in the near future that would close these gaps in the available information.

Comparison of this data to other available LULC characterizations, particularly the Major Land Use (MLU) data and the Cropland Data Layer (CDL) from the USDA, would provide useful validation metrics.

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.

European Commission, Joint Research Centre (2003). Global land cover 2000 database.

Fisher, P. F., A. J. Comber, 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. A. (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.

Hijmans, R. J. and J. van Etten (2011). *raster: Geographic analysis and modeling with raster data*. R package version 1.8-12.

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.

LP DAAC (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).

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/.

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.

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.

Appendix: Source Code

PEEL processing library

```
library( raster)
library( ggplot2)
library( xtable)
library( RColorBrewer)
library( Hmisc)

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
}


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
  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$Ap <- calc( stack(mlct$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
}

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
}


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]
}
```

```
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))))
        })
}
```

Map rendering library

```
theme_set( theme_bw( base_family= "serif"))

theme_update( panel.grid.minor= theme_blank(),
              panel.grid.major= theme_blank(),
              panel.background= theme_blank(),
              axis.title.x= theme_blank(),
              axis.text.x= theme_text( family= "serif",
                angle= 90, hjust= 1 ),
              axis.text.x= theme_text( family= "serif"),
              axis.title.y= theme_blank())

theme_map <- theme_get()

theme_set( theme_bw( base_family= "serif"))

ggplotRaster <- function( r, samp) {
  df <- data.frame( as( sampleRegular( r, ncell( r)*samp,
                                       asRaster=TRUE),
                        "SpatialGridDataFrame"))
  ext <- extent( r)
```

```
  ggplot( data= df) +
    geom_tile( aes( x= s1, y= s2, fill= values)) +
    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)) +
    theme_map +
    coord_equal()
}


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


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

```
                        "SpatialGridDataFrame"))
  names( df)[ grep( "^values", names( df))] <- layerNames( r)
  df <- df[, c( "s1", "s2", classes)]
  df <- melt( df, id.vars= c("s1", "s2"))
  ggplot( data= df) +
    geom_tile( aes( x= s1, y= s2, fill= value)) +
    scale_x_continuous( expand= c( 0,0)) +
    scale_y_continuous( expand= c( 0,0)) +
    scale_fill_gradientn( colours= rev( brewer.pal( 6, "YlGn")),
                          limits= c( 1, 0),
                          breaks= seq( 1, 0, by= -0.2)) +
    facet_wrap(~ variable) +
    theme_map +
    coord_equal()
}

my.ggsave <- function( wd= getwd(),
                       filename= default_name(plot),
                       height= 5, width= 5, ...) {
  ggsave(filename= paste( wd, filename, sep= "/"),
         height= height,
         width= width, ...)
}

coverDiffMaps <-
  function( r, samp= 1,
            classes= layerNames( r),
            breaks= c( -1, -0.5, -0.1, -0.05, -0.01,
                       0.01, 0.05, 0.1, 0.5, 1),
            ...) {
  df <- data.frame( as( sampleRegular( r, ncell( r)*samp,
                                       asRaster=TRUE),
```

```
                    "SpatialGridDataFrame"))
  names( df)[ grep( "^values", names( df))] <- layerNames( r)
  df <- df[, c( "s1", "s2", classes)]
  df <- melt( df, id.vars= c("s1", "s2"))
  df <- within( df,
              cuts <- cut( value,
                           breaks= breaks,
                           include.lowest= TRUE))
  pal <- brewer.pal( length( levels( df$cuts)), "BrBG")
  names( pal) <- levels( df$cuts)
  ggplot( data= df) +
    geom_tile( aes( x= s1, y= s2, fill= cuts)) +
    scale_fill_manual( "offset", values= pal, breaks= rev( names( pal))) +
    facet_grid( variable ~ .) +
    scale_x_continuous( expand= c( 0,0)) +
    scale_y_continuous( expand= c( 0,0)) +
    theme_map
}
```

Code from Chapter 2

```
################################################
### chunk number 1: initialize
################################################
#line 17 "/home/nbest/thesis/datasets.Rnw"


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


overwriteRasters <- FALSE
```

```
overwriteFigures <- TRUE



##################################################
### chunk number 2: thumb
##################################################
#line 102 "/home/nbest/thesis/datasets.Rnw"

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


## 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

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",
##                 "#00A25B",
##                 "#00A125",
##                 "#069228",
##                 "#9E9668",
##                 "#C1C48F",
##                 "#85AA5B",
##                 "#B1B741",
##                 "#A4D07E",
##                 "#73ABAE",
##                 "#CCD253",
##                 "#D90000",
##                 "#9DE36E",
##                 "#B6B5C2",
##                 "#949494")
```

```
#####################################################
### chunk number 3: mlct-reclass
#####################################################
#line 202 "/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( "%␣conf",
```

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



##################################################
### chunk number 4: fig_thumb_pri_reclass
##################################################
#line 224 "/home/nbest/thesis/datasets.Rnw"

if( overwriteFigures) {
  my.ggsave( texWd, "fig_thumb_pri_reclass.png",
            plot= thumbPlots$pri)
}



##################################################
### chunk number 5: fig_thumb_sec_reclass
##################################################
#line 250 "/home/nbest/thesis/datasets.Rnw"


if( overwriteFigures) {
  my.ggsave( texWd, "fig_thumb_sec_reclass.png",
          plot= thumbPlots$sec)
}
```

```
#################################################
### chunk number 6: fig_thumb_pct
#################################################
#line 282 "/home/nbest/thesis/datasets.Rnw"

if( overwriteFigures) {
  my.ggsave( texWd, "fig_thumb_pct.png",
             plot= thumbPlots$pct)
}



#################################################
### chunk number 7: fig_thumb_pri_facet
#################################################
#line 320 "/home/nbest/thesis/datasets.Rnw"

if( overwriteFigures) {
  my.ggsave( texWd, "fig_thumb_pri_facet.png",
             plot= thumbPlots$pri +
               facet_wrap(~ values) +
               opts( legend.position= "none"))
}



#################################################
### chunk number 8: fig_thumb_sec_facet
#################################################
#line 341 "/home/nbest/thesis/datasets.Rnw"

if( overwriteFigures) {
```

```
  my.ggsave( texWd, "fig_thumb_sec_facet.png",
            plot= thumbPlots$sec +
              facet_wrap(~ values) +
              opts( legend.position= "none"))
}



##################################################
### chunk number 9: mlct_reclass
##################################################
#line 382 "/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$pri, 16e-4),
                     sec= peelMap( mlct$sec, 16e-4))
  mlctPlots$pct <- ggplotRaster( mlct$pct, 16e-4) +
    scale_fill_gradientn( "% conf",
                          colours= rev( brewer.pal( 7, "YlGn")),
                          limits= c( 100, 0),
                          breaks= seq( 100, 0, by= -20))
}
```

```
##################################################
### chunk number 10: fig_mlct_pri_reclass
##################################################
#line 408 "/home/nbest/thesis/datasets.Rnw"

if( overwriteFigures) {
  my.ggsave( texWd, "fig_mlct_pri_reclass.png",
            plot= mlctPlots$pri, width=7.5)
  system( sprintf( "convert␣-trim␣%s/fig_mlct_pri_reclass.png␣%s/fig_mlct_pri_
     reclass_trim.png",
                texWd, texWd))
}


##################################################
### chunk number 11: fig_mlct_sec_reclass
##################################################
#line 427 "/home/nbest/thesis/datasets.Rnw"

if( overwriteFigures) {
  my.ggsave( texWd, "fig_mlct_sec_reclass.png",
            plot= mlctPlots$sec, width=7.5)
  system( sprintf( "convert␣-trim␣%s/fig_mlct_sec_reclass.png␣%s/fig_mlct_sec_
     reclass_trim.png",
                texWd, texWd))
}


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

```
### chunk number 12: fig_mlct_pct
####################################################
#line 447 "/home/nbest/thesis/datasets.Rnw"

if( overwriteFigures) {
  my.ggsave( texWd, "fig_mlct_pct.png",
           plot= mlctPlots$pct, width=7.5)
  system( sprintf( "convert␣-trim␣%s/fig_mlct_pct.png␣%s/fig_mlct_pct_trim.png
     ",
                  texWd, texWd))
}



####################################################
### chunk number 13: fig_mlct_pri_facet
####################################################
#line 467 "/home/nbest/thesis/datasets.Rnw"

if( overwriteFigures) {
  my.ggsave( texWd, "fig_mlct_pri_facet.png",
         plot= peelMap( mlct$pri, 16e-4,
                       classes= names( peelClasses)[1:5]) +
           facet_grid( values ~ .) +
           opts( legend.position= "none"),
         width=4.5, height=8)
}



####################################################
### chunk number 14: fig_mlct_pri_facet2
```

```
##################################################
#line 488 "/home/nbest/thesis/datasets.Rnw"

if( overwriteFigures) {
  my.ggsave( texWd, "fig_mlct_pri_facet2.png",
          plot= peelMap( mlct$pri, 16e-4,
                        classes= names( peelClasses)[6:9]) +
            facet_grid( values ~ .) +
            opts( legend.position= "none"),
          width=4.5, height=8)
}



##################################################
### chunk number 15: fig_mlct_sec_facet
##################################################
#line 510 "/home/nbest/thesis/datasets.Rnw"

if( overwriteFigures) {
  my.ggsave( texWd, "fig_mlct_sec_facet.png",
             plot= peelMap( mlct$sec, 16e-4,
                        classes= names( peelClasses)[1:5]) +
               facet_grid( values ~ .) +
               opts( legend.position= "none"),
             width=4.5, height=8)
}



##################################################
### chunk number 16: fig_mlct_sec_facet2
```

```
###################################################
#line 531 "/home/nbest/thesis/datasets.Rnw"

if( overwriteFigures) {
  my.ggsave( texWd, "fig_mlct_sec_facet2.png",
             plot= peelMap( mlct$sec, 8e-3,
                        classes= names( peelClasses)[6:9]) +
               facet_grid( values ~ .) +
               opts( legend.position= "none"),
             width=4.5, height=8)
}



###################################################
### chunk number 17: thumbPlots
###################################################
#line 607 "/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")


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 18: fig_thumb_fracs
##################################################
#line 651 "/home/nbest/thesis/datasets.Rnw"

if( overwriteFigures) {
  my.ggsave( texWd, "fig_thumb_fracs.png",
             plot= thumbPlots$fracs)
}



##################################################
### chunk number 19: fig_thumb1_fracs
##################################################
#line 676 "/home/nbest/thesis/datasets.Rnw"
```

```
if( overwriteFigures) {
  my.ggsave( texWd, "fig_thumb1_fracs.png",
            plot= thumbPlots1$fracs)
}
```

```
#####################################################
### chunk number 20: fig_thumb1_agg
#####################################################
#line 708 "/home/nbest/thesis/datasets.Rnw"

if( overwriteFigures) {
  my.ggsave( texWd, "fig_thumb1_agg.png",
            plot= thumbPlots1$agg)
}
```

```
#####################################################
### chunk number 21: fig_thumb_agg
#####################################################
#line 726 "/home/nbest/thesis/datasets.Rnw"

if( overwriteFigures) {
  my.ggsave( texWd, "fig_thumb_agg.png",
            plot= thumbPlots$agg)
}
```

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

```
### chunk number 22: thumbAggDiff
################################################
#line 755 "/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 23: fig_thumb_agg_diff
################################################
#line 793 "/home/nbest/thesis/datasets.Rnw"

if( overwriteFigures) {
  my.ggsave( texWd, "fig_thumb_agg_diff.png",
             plot= thumbAggDiffPlot)
```

```
}


##################################################
### chunk number 24: mlct_agg
##################################################
#line 824 "/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")

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

```
#################################################
### chunk number 25: thumbNomos
#################################################
#line 913 "/home/nbest/thesis/datasets.Rnw"

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

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

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

if( overwriteFigures) {
  thumbNomosDiffPlot <- coverMaps( thumbNomosDiff) +
    scale_fill_gradientn( "diff", colours= rev( brewer.pal( 11, "BrBG")),
                          limits= c( 0.3, -0.3),
                          breaks= seq( 0.3, -0.3, by= -0.06))
}
```

```
##################################################
### chunk number 26: fig_thumb1_nomos
##################################################
#line 945 "/home/nbest/thesis/datasets.Rnw"

if( overwriteFigures) {
  my.ggsave( texWd, "fig_thumb1_nomos.png",
            plot= thumbPlots1$nomos)
}



##################################################
### chunk number 27: fig_thumb_nomos
##################################################
#line 964 "/home/nbest/thesis/datasets.Rnw"

if( overwriteFigures) {
  my.ggsave( texWd, "fig_thumb_nomos.png",
            plot= thumbPlots$nomos)
}



##################################################
### chunk number 28: fig_thumb_nomos_diff
##################################################
#line 986 "/home/nbest/thesis/datasets.Rnw"
```

```
if( overwriteFigures ) {
  my.ggsave( texWd, "fig_thumb_nomos_diff.png",
         plot= thumbNomosDiffPlot)
}
```

```
#################################################
### chunk number 29: mlct_nomos
#################################################
#line 1004 "/home/nbest/thesis/datasets.Rnw"

setwd( dataWd)

mlct  <- decomposeMosaic( mlct, 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 30: thumb_nlcd
#################################################
#line 1078 "/home/nbest/thesis/datasets.Rnw"

setwd( dataWd)
```

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

thumbNlcd <- list( pri=raster( "thumbNlcd.tif"))



##################################################
### chunk number 31: thumb_nlcd_reclass
##################################################
#line 1157 "/home/nbest/thesis/datasets.Rnw"

setwd( nlcdWd)


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

if( overwriteFigures) {
  thumbNlcdPlot <- peelMap(thumbNlcd$pri, 0.1)
}



##################################################
### chunk number 32: fig_thumb_nlcd_reclass
##################################################
#line 1176 "/home/nbest/thesis/datasets.Rnw"

if( overwriteFigures) {
  my.ggsave( texWd, "fig_thumb_nlcd_reclass.png",
```

```
            plot= thumbNlcdPlot, height= 5, width= 5)
}



##################################################
### chunk number 33: fig_thumb_nlcd_facet
##################################################
#line 1194 "/home/nbest/thesis/datasets.Rnw"

if( overwriteFigures) {
  my.ggsave( texWd, "fig_thumb_nlcd_facet.png",
            plot= thumbNlcdPlot +
              facet_wrap(~ values) +
              opts( legend.position= "none"))
}



##################################################
### chunk number 34: thumb_nlcd_aggr
##################################################
#line 1224 "/home/nbest/thesis/datasets.Rnw"

setwd( nlcdWd)

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)
}



##################################################
### chunk number 35: fig_thumb_nlcd_agg
##################################################
#line 1247 "/home/nbest/thesis/datasets.Rnw"

if( overwriteFigures) {
  my.ggsave( texWd, "fig_thumb_nlcd_agg.png",
             plot= thumbNlcdAggPlot)
}



##################################################
### chunk number 36: nlcd
##################################################
#line 1265 "/home/nbest/thesis/datasets.Rnw"

setwd( dataWd)

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])
```

```
##################################################
### chunk number 37: fig_nlcd
##################################################
#line 1285 "/home/nbest/thesis/datasets.Rnw"

if( overwriteFigures) {
  nlcdPlot <- coverMaps( nlcd, samp= 0.2,
                        classes= layerNames( nlcd)[ 1:4]) +
    facet_grid( variable ~ .)
  my.ggsave( texWd, "fig_nlcd.png", width=5.5, height=8)
}
```

```
##################################################
### chunk number 38: fig_nlcd2
##################################################
#line 1303 "/home/nbest/thesis/datasets.Rnw"

if( overwriteFigures) {
  nlcdPlot2 <- coverMaps( nlcd, 0.2,
```

```
                        classes= layerNames( nlcd)[ 5:8]) +
    facet_grid( variable ~ .)
  my.ggsave( texWd, "fig_nlcd2.png", width=5.5, height=8)
}



##################################################
### chunk number 39: agland
##################################################
#line 1347 "/home/nbest/thesis/datasets.Rnw"

setwd( dataWd)
setwd( "agland")

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

```
                                        # crop() returns a brick


if( overwriteFigures) {
  thumbAglandPlot <-
    coverMaps( thumbAgland, 1) +
      facet_grid( variable ~ .)

  aglandPlot <-
    coverMaps( agland, 0.4) +
      facet_grid( variable ~ .)

}




##################################################
### chunk number 40: fig_thumb_agland
##################################################
#line 1390 "/home/nbest/thesis/datasets.Rnw"

if( overwriteFigures) {
  my.ggsave( texWd, "fig_thumb_agland.png",
            plot= thumbAglandPlot)
}




##################################################
### chunk number 41: fig_agland
##################################################
```

```
#line 1409 "/home/nbest/thesis/datasets.Rnw"

if( overwriteFigures) {
  my.ggsave( texWd, "fig_agland.png",
         plot= aglandPlot)
  system( sprintf( "convert␣-trim␣%s/fig_agland.png␣%s/fig_agland_trim.png",
                  texWd, texWd))
}



##################################################
### chunk number 42: 175crops
##################################################
#line 1453 "/home/nbest/thesis/datasets.Rnw"

setwd( dataWd)

cropsWd <- path.expand( "~/see/data/raw/175crops2000/nc")
## list.files( cropsWd, "vrt$")

cropTable <- read.csv( "monfreda2008␣table1.csv", header= TRUE)

## For now we consider only herbaceous crops


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( "mask_cusa.tif")
```

```
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)
  raster::mask( r, cusaMask,
       filename= paste( c, "tif", sep="."),
       overwrite= TRUE)
})
names( catMasked) <- names( catStacks)




################################################
### chunk number 43: fig_crops
################################################
#line 1567 "/home/nbest/thesis/datasets.Rnw"

if( overwriteFigures) {
  cropsMap <-
```

```
    coverMaps( stack( catMasked[ 1:5]), 0.4) +
    facet_grid( variable ~ .)
  my.ggsave( texWd, "fig_crops.png",
            width=5.5, height=8)
}
```

```
#################################################
### chunk number 44: fig_crops2
#################################################
#line 1589 "/home/nbest/thesis/datasets.Rnw"

if( overwriteFigures) {
  cropsMap2 <-
    coverMaps( stack( catMasked[ 6:10]), 0.4) +
    facet_grid( variable ~ .)
  my.ggsave( texWd, "fig_crops2.png",
            width=5.5, height=8)
}
```

Code from Chapter 3

```
#################################################
### chunk number 1: init
#################################################
#line 21 "/home/nbest/thesis/analysis.Rnw"

options( prompt= "␣", continue= "␣", width= 60)
options(error= function(){
  ## recover()
  options( prompt= ">␣", continue= "+␣", width= 80)
```

```
})

source( "~/thesis/code/peel.R")
source( "~/thesis/code/maps.R")

   texWd <- path.expand( "~/thesis/analysis")
rasterWd <- path.expand( "~/thesis/data/analysis")
dataPath <- path.expand( "~/thesis/data")
setwd( rasterWd)

overwriteRasters <- TRUE
overwriteFigures <- TRUE

                                        # studyArea used to work out RMSE
                                        # calcs and tables
##studyArea <- "thumb"
studyArea <- "mlct"
                                        # bands are numbered from one but
                                        # classes from zero.  Used for stacks/
                                           brick
                                        # where bands correspond to classes
peelBands <- peelClasses +1

                                        # mask and agland exported from GRASS
                                        # no need to mask or crop
cusaMask <- raster( sprintf( "%s/mask_cusa.tif",
                            dataPath))
cusaExtent <- extent( cusaMask)
thumbExtent <- extent( -( 83 +30 /60), -( 82 +25 /60),
                          42 +55 /60,     44  +5 /60 )

                                        # default raster() output
```

```
                                        # has geographic proj, full extent
                                        # by default
world <- raster()
res(world) <- 5/60

grid <- raster( cusaMask)
grid[] <- cellsFromExtent( world, grid)
grid <- raster::mask( grid, cusaMask)

nulls <- raster( cusaMask)
nulls[] <- NA

zeroes <- raster( cusaMask)
zeroes[] <- 0

ones <- raster( cusaMask)
ones[] <- 1

if( studyArea == "thumb") {
  cusaMask <- crop( cusaMask, thumbExtent)
}
acresFile <- paste( "acres",
                    paste( studyArea, ".tif", sep=""),
                    sep="_")
if( overwriteRasters) {
  acres <- area( cusaMask) *247.105381
  acres <- writeRaster( acres,
                        filename= acresFile,
                        overwrite= TRUE)
} else acres <- raster( acresFile)

agland <- stack( list.files( paste( dataPath, "agland", sep="/"),
```

```
                              patt= "(cropland|pasture).tif$",
                              full.names= TRUE))
layerNames(agland) <- c("crop", "open")
agland <- setMinMax( agland)
if( studyArea == "thumb") {
  agland <- crop( agland, thumbExtent)
}

agg05 <-
  brick( list.files( dataPath,
                     patt= paste( studyArea, "_Amin_0.5_agg.tif", sep=""),
                     full.names= TRUE))
layerNames( agg05) <- names( peelClasses)

nomos05 <-
  brick( list.files( dataPath,
                     patt= paste( studyArea, "_Amin_0.5_nomosaic.tif", sep=""),
                     full.names= TRUE))
layerNames( nomos05) <- c( names( peelClasses)[ -8], "total")

agg1 <-
  brick( list.files( dataPath,
                     patt= paste( studyArea, "1_agg.tif", sep=""),
                     full.names= TRUE))
layerNames( agg1) <- names( peelClasses)

nomos1 <-
  brick( list.files( dataPath,
                     patt= paste( studyArea, "1_Amin_1_nomosaic.tif", sep=""),
                     full.names= TRUE))
layerNames( nomos1) <- c( names( peelClasses)[ -8], "total")
```

```
nlcd <-
  brick( sapply( names( peelClasses),
                 function( cover) {
                   if( cover == "mosaic") {
                     zeroes
                   } else {
                     fn <-
                       list.files( paste( dataPath, "nlcd",
                                          sep = "/"),
                                   patt= paste( "nlcd", cover, "5min.tif$",
                                                sep = "_"),
                                   full.names= TRUE)
                     crop( raster( fn), cusaMask)
                   }}))


nlcd <- writeRaster( nlcd,
                     filename= paste( path.expand(rasterWd),
                       "nlcd.tif",
                       sep= "/"),
                     overwrite= TRUE)

layerNames( nlcd) <- names( peelClasses)


rasterNames <- c( "agland", "nlcd", "agg05", "agg1", "nomos05", "nomos1")

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)))
  })
areasDf <-
  areasDf[, c( 1, 3, 4)]
colnames( areasDf) <-
  c( "map", "class", "acres")
areasDf$map <-
  factor( areasDf$map,
          levels= rasterNames)

legendOrder <- rev( c( 6, 4, 2, 3, 9, 7, 5, 1, 8))

areasDf$class <-
  factor( areasDf$class,
          levels= c( names( peelLegend)[ rev( legendOrder)], "total"))

if( overwriteFigures) areasPlot <-
  qplot( map, acres /10^6,
         data= subset(areasDf, class != "total"),
         geom="bar", position= "stack",
         fill= class,
         stat="summary", fun.y="sum") +
  scale_fill_manual( "",
                     values= peelLegend[ legendOrder],
                     ##peelLegend[ levels( areasDf$class)[1:9]],
```

```
                    breaks= names( peelLegend)[ legendOrder]) +
  scale_y_continuous( "M␣acres",
      limits= c(0,2000)) +
  theme_bw( base_family= "serif") +
  scale_x_discrete( "",
      limits= rasterNames[ c( 1, 2, 4, 3, 6, 5)],
      breaks= rasterNames[ c( 1, 2, 4, 3, 6, 5)],
      labels= expression("Agland2000", "NLCD",
          atop( atop( textstyle( "MLCT"),
                      textstyle( A[ min] ==1.0)),
               phantom(0)),
          atop( atop( textstyle( "MLCT"),
                      textstyle( A[ min] ==0.5)),
               phantom(0)),
          atop( atop( textstyle( "MLCT"),
                      textstyle( A[ min] ==1.0)),
               "No␣Mosaic"),
          atop( atop( textstyle( "MLCT"),
                      textstyle( A[ min] ==0.5)),
               "No␣Mosaic")))

areasCt <- cast( areasDf, class ~ map,
                 value= "acres",
                 subset= class != "total",
                 sum,
                 margins="grand_row")[, -1]
rownames( areasCt) <- levels( areasDf$class)



################################################
### chunk number 2: tab_areas
```

```
###################################################
#line 308 "/home/nbest/thesis/analysis.Rnw"


local({
  colnames( areasCt) <- c( "Agland2000", "NLCD",
                           "\\pbox[c][][c]{3in}{Aggregated\\\\$A_{min}=0.5$}",
                           "\\pbox[c][][c]{3in}{Aggregated\\\\$A_{min}=1.0$}",
                           "\\pbox[c][][c]{3in}{No␣Mosaic\\\\$A_{min}=0.5$}",
                           "\\smallskip\\pbox[c][][c]{3in}{No␣Mosaic\\\\$A_{min
                              }=1.0$}")
  print( xtable( areasCt / 10^6,
                 caption= "Total␣Acreages␣by␣Map␣and␣Cover",
                 label= "tab:areas",
                 digits= 1),
         add.to.row= list(
           pos= list( 0, nrow( areasCt)),
           command= rep("\\noalign{\\smallskip}", times= 2)),
         size= "small",
         sanitize.colnames.function= function(x) x)
})




###################################################
### chunk number 3: fig_areas
###################################################
#line 335 "/home/nbest/thesis/analysis.Rnw"

if( overwriteFigures) {
```

```
  setwd( texWd)
  my.ggsave( texWd, "fig_areas.pdf",
         device= pdf,
         plot= areasPlot,
         width= 6,
         height=6)
}
```

```
#####################################################
### chunk number 4: nomosDiff
#####################################################
#line 387 "/home/nbest/thesis/analysis.Rnw"

if( overwriteFigures) {
  nomosDiff1 <- getPeelBand( nomos1, "crop") -aglandCrop
  layerNames( nomosDiff1) <- "crop"
  nomosDiffPlot1 <- coverMaps( nomosDiff1, classes= "crop", samp= 0.2) +
    scale_fill_gradientn( "diff", colours= rev( brewer.pal( 11, "BrBG")),
                          limits= c( 1, -1),
                          breaks= seq( 1, -1, by= -0.2))
  nomosDiff05 <- getPeelBand( nomos05, "crop") -aglandCrop
  layerNames( nomosDiff05) <- "crop"
  nomosDiffPlot05 <- coverMaps( nomosDiff05, classes= "crop", samp= 0.2) +
    scale_fill_gradientn( "diff", colours= rev( brewer.pal( 11, "BrBG")),
                          limits= c( 1, -1),
                          breaks= seq( 1, -1, by= -0.2))
}
```

```
#####################################################
### chunk number 5: fig_nomosDiff1
#####################################################
#line 409 "/home/nbest/thesis/analysis.Rnw"
if( overwriteFigures) {
  my.ggsave( texWd, "fig_nomosDiff1.png", plot= nomosDiffPlot1)
}


#####################################################
### chunk number 6: fig_nomosDiff05
#####################################################
#line 423 "/home/nbest/thesis/analysis.Rnw"
if( overwriteFigures) {
  my.ggsave( texWd, "fig_nomosDiff05.png", plot= nomosDiffPlot05)
}


#####################################################
### chunk number 7: rmse
#####################################################
#line 450 "/home/nbest/thesis/analysis.Rnw"


rmseDf <- ldply( list("nomos05", "nomos1"),
                function( brickName) {
                  rmseRast( getPeelBand( get( brickName), "crop"),
                           aglandCrop)
                })
rmseDf <- cbind( c( 0.5, 1.0), rmseDf)
colnames( rmseDf) <- c( "$A_{min}$", "RMSE")
```

```
cropScatDf <-
  data.frame( as( stack( getPeelBand( nomos05, "crop"),
                         getPeelBand( nomos1, "crop"),
                         aglandCrop,
                         raster::mask(acres, cusaMask)),
                 "SpatialGridDataFrame"))
colnames(cropScatDf) <-
  c( "nomos05", "nomos1", "agland", "acres", "lon", "lat")
cropScatDf$weight <- with( cropScatDf, acres/ max(acres))



if( overwriteFigures) hexPlot1 <-
  ggplot( data= cropScatDf,
       aes( agland, nomos1)) +
  stat_binhex( binwidth= c( 0.025, 0.025)) +
  scale_fill_gradientn( colours= brewer.pal( 6, "YlGn"),
                        trans= "log10",
                        limits=c( 10, 10000)) +
  geom_abline( alpha=0.4) +
  scale_x_continuous( "Agland2000",
                      expand= c( 0,0.0125)) +
  scale_y_continuous( expression( paste("MLCT, ", A[min] == 1.0)),
                      expand= c( 0,0.0125)) +
  theme_bw( base_family= "serif") +
  coord_equal() +
  opts( panel.grid.minor= theme_blank(),
        panel.grid.major= theme_blank(),
        panel.background= theme_blank())
```

```
#################################################
### chunk number 8: fig_hexplot1
#################################################
#line 496 "/home/nbest/thesis/analysis.Rnw"

if( overwriteFigures) {
  my.ggsave( texWd, "fig_hexPlot1.pdf",
          dev= pdf,
          plot= hexPlot1)
          ## width= 4.5,
          ## height= 4.5)
}



#################################################
### chunk number 9: fig_hexplot05
#################################################
#line 550 "/home/nbest/thesis/analysis.Rnw"

if( overwriteFigures) {
  my.ggsave( texWd, "fig_hexPlot05.pdf",
          device= pdf,
          plot= hexPlot1 +
             aes(agland, nomos05) +
             scale_y_continuous( expression(paste("MLCT, ", A[min] == 0.5)),
                                 limits= c( 0, 1),
                                 breaks= seq( 0, 1, by= 0.2),
                                 expand= c( 0,0.0125)))
  }
```

```
##################################################
### chunk number 10: table_rmse
##################################################
#line 574 "/home/nbest/thesis/analysis.Rnw"

print( xtable( rmseDf,
               caption= "RMSE, MLCT vs. Agland2000 crop",
               label= "tab:rmse",
               digits= c( 0, 1, 3)),
      include.rownames= FALSE,
      sanitize.colnames.function= function(x) x)



##################################################
### chunk number 11: offsets_calc
##################################################
#line 666 "/home/nbest/thesis/analysis.Rnw"

nlcdKeep <- stack( llply( names( peelClasses), function( class) {
  if( class %in% c( "water", "wetland", "urban"))
    ones else zeroes
}))

nlcdIgnore <- stack( llply( names( peelClasses), function( class) {
  if( class %in% c( "water", "wetland", "urban"))
    zeroes else ones
}))
```

```
nlcdKeepOffsets <-
  (nlcd -agg05) *nlcdKeep

mlctKeep <- agg05 *nlcdIgnore

nlcdIgnoreOffsets <-
  overlay( mlctKeep, sum( mlctKeep), sum( nlcdKeepOffsets),
          fun= function( mk, smk, snko) {
            ifelse( mk == 0 & smk ==0,
                    0,
                    -1 *mk /smk *snko)
          })


nlcdOffsets <- nlcdKeepOffsets +nlcdIgnoreOffsets

nlcdOffsets <-
  writeRaster( nlcdOffsets,
               filename= paste( rasterWd, "nlcdOffsets.tif", sep= "/"),
               overwrite= TRUE)


nlcdOffsets <- stack( nlcdOffsets, sum( nlcdOffsets))
layerNames( nlcdOffsets) <- c( names( peelClasses), "total")

thumbNlcdOffsets <- crop( nlcdOffsets, thumbExtent)

offsetsMap1 <- coverDiffMaps( nlcdOffsets, samp= 0.4,
                              classes= layerNames( nlcdOffsets)[ 1:5]) +
               coord_equal()
```

```
offsetsMap2 <- coverDiffMaps( nlcdOffsets, samp= 0.4,
                              classes= layerNames( nlcdOffsets)[ 6:10]) +
               coord_equal()



thumbOffsetsMap <-
  coverDiffMaps( thumbNlcdOffsets,
                 classes= layerNames( thumbNlcdOffsets)[-10]) +
  facet_wrap( ~variable)




##################################################
### chunk number 12: fig_offsetsmap1
##################################################
#line 727 "/home/nbest/thesis/analysis.Rnw"

if( overwriteFigures) {
  my.ggsave( texWd, "fig_offsets1.png", plot= offsetsMap1, height= 7)
}



##################################################
### chunk number 13: fig_offsets2
##################################################
#line 745 "/home/nbest/thesis/analysis.Rnw"

if( overwriteFigures) {
  my.ggsave( texWd, "fig_offsets2.png", plot= offsetsMap2, height= 7)
```

```
}
```

```
##################################################
### chunk number 14: cor_offsets
##################################################
#line 773 "/home/nbest/thesis/analysis.Rnw"

corOffsets <- cor( data.frame(as( nlcdOffsets, "SpatialGridDataFrame"))[,
   1:9],
                  use= "complete.obs")
colnames( corOffsets) <- names( peelClasses)
rownames( corOffsets) <- names( peelClasses)

corOffsetsPlot <-
  ggplot( melt( corOffsets),
         aes( x=X1, y=X2, fill= value)) +
  geom_tile() +
  theme_bw( base_family= "serif") +
  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( corOffsets)) +
  scale_y_discrete( limits= colnames( corOffsets)) +
  scale_fill_gradientn( "", colours= rev( brewer.pal( 11, "BrBG")),
                       limits= c( 1.0, -1.0),
                       breaks= seq( 1.0, -1.0, by= -0.2))
```

```
if( overwriteFigures) {
  oldWd <- setwd( texWd)
  ggsave( "fig_corOffsets.pdf",
          device= pdf,
          plot= corOffsetsPlot,
          height= 4.5,
          width= 4.5)
  setwd( oldWd)
}
```

```
##################################################
### chunk number 15: cusa_offset
##################################################
#line 857 "/home/nbest/thesis/analysis.Rnw"

setwd( rasterWd)

                                        # reload offsets to get rid
                                        # of total layer
nlcdOffsets <- brick( paste( rasterWd, "nlcdOffsets.tif", sep="/"))
layerNames( nlcdOffsets) <- names( peelClasses)

mlctAdj <- list( Amin=0.5)
mlctAdj$agg <-
  if( overwriteRasters) {
    overlay( agg05, nlcdOffsets,
            fun= sum,
            filename= "agg05Adj.tif",
```

```
            overwrite= TRUE)
  } else brick( list.files( rasterWd,
                           patt= "agg05Adj.tif",
                           full.names= TRUE))

layerNames( mlctAdj$agg) <- names( peelClasses)

mlctAdj  <- decomposeMosaic( mlctAdj, overwrite= overwriteRasters, progress= "
    text")



##################################################
### chunk number 16: areas2
##################################################
#line 887 "/home/nbest/thesis/analysis.Rnw"

# reuse area table code from above; better to implement a function?

rasterNames2 <- c( "agland", "nlcd", "agg05", "nomos05",
                  "nlcdOffsets", "mlctAdj$agg", "mlctAdj$nomos")

dataSets2 <- sapply( rasterNames2,
  function( n) eval( parse( text=n)))

areas2 <- llply( dataSets2,
  function( d) {
    res <- cellStats( d *acres, sum)
    names( res) <- layerNames( d)
    res
  })
```

```
areasDf2 <- ldply( areas2, function( a) {
  melt( t( as.data.frame( a)))
})
areasDf2 <-
  areasDf2[, c( 1, 3, 4)]
colnames( areasDf2) <-
  c( "map", "class", "acres")

areasDf2 <-
  transform( areasDf2,
            class= factor( class,
              levels= c( names( peelLegend)[ rev( legendOrder)], "total")),
              ## c("crop", "open",
              ##   names( peelClasses)[-c(4,6,8)],
              ##   "mosaic", "total")),
            map= factor( map,
              levels= rasterNames2))

areasCt2 <- cast( areasDf2,
                 class ~ map,
                 subset= class != "total",
                 value= "acres",
                 sum,
                 margins="grand_row")

rownames( areasCt2) <- areasCt2[, "class"]
areasCt2 <- areasCt2[, -1]
areasCt2 <- areasCt2[ c( names( peelClasses), "(all)"), rasterNames2]



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

```
### chunk number 17: restack_check
################################################
#line 936 "/home/nbest/thesis/analysis.Rnw"

## check that everything balances
## output of decomposeMosaic is not brick()ed properly
## in the sense that the layer set is incomplete
## and out of order


restack <- function( peelBrick) {
  u <- unstack( peelBrick)
  names( u) <- layerNames( peelBrick)
  r <- do.call( stack,
          llply( names( peelClasses),
                 function( cover) {
                   if( is.null( u[[ cover]]))
                     zeroes
                   else
                     u[[ cover]]
                 }))
  layerNames( r) <- names( peelClasses)
  r
}
                                          # restack() takes any of the bricks/
                                             stacks from
                                          # previous functions and rearranges
                                             the layers
                                          # to match the PEEL classes, inserting
                                              layers of
                                          # zeroes as needed
```

```
restackOverlay <- function( rasterList, fun) {
  l <- llply( rasterList, restack)
  names( l) <- NULL
  do.call( overlay, c( l, fun=fun))
}
                                        # restackOverlay() runs its arguments
                                            through restack()
                                        # and applies a function to its
                                            outputs



##################################################
### chunk number 18: table_restack_check eval=FALSE
##################################################
## #line 978 "/home/nbest/thesis/analysis.Rnw"
##
## check <- restackOverlay( c( mlctAdj[ c("nomos", "delta")],
##                            nlcdOffsets,
##                            agg05),
##                         function( n, d, o, a) n-d-o-a)
## layerNames(check) <- names( peelClasses)
##
## checkTable <-
##   xtable( cbind( class=peelClasses,
##                 min=minValue( check),
##                 max=maxValue(check)),
##          caption= "Balance of adjustment fractions and original MLCT
   aggregation",
##          label= "tab:restack_check")
## digits( checkTable) <- c( 0, 0,-2,-2)
```

```
## print( checkTable)
##


################################################
### chunk number 19: table_rmse2
################################################
#line 1003 "/home/nbest/thesis/analysis.Rnw"

                                        # add the RMSE for the new crop map
                                        # and an indication of the NLCD
                                            offsets' presence

rmseDf2 <-
  cbind( offset=c( TRUE, FALSE, FALSE),
        rbind( c( 0.5,
                  rmseRast( getPeelBand( mlctAdj$nomos, "crop"),
                            aglandCrop)),
              rmseDf))

##                                         # add the RMSE for the open class
## rmseDf2 <-
##   cbind( rmseDf2,
##         rmseOpen=ldply( list(mlctAdj$nomos, nomos05, nomos1),
##                 function( brickVar) {
##                   rmseRast( getPeelBand( brickVar, "open"),
##                             unstack( agland)[[ 2]])
##                 }))
## colnames(rmseDf2)[ c(3,4)] <- c( "$RMSE_{crop}$", "$RMSE_{open}$")


print( xtable( rmseDf2,
```

```
            caption= "RMSE, MLCT vs. Agland2000 crop with NLCD offsets",
            label= "tab:rmse2",
            digits= c( 0, 0, 1, 3)),
      include.rownames= FALSE,
      sanitize.colnames.function= function(x) x)



##################################################
### chunk number 20: tab_areas2
##################################################
#line 1048 "/home/nbest/thesis/analysis.Rnw"


local({
  colnames( areasCt2) <- c( "Agland2000", "NLCD", "MLCT",
                           "\\pbox[c][][c]{3in}{MLCT\\\\No Mosaic}",
                           "\\pbox[c][][c]{3in}{NLCD\\\\Offsets}",
                           "\\pbox[c][][c]{3in}{MLCT\\\\Adjusted}",
                           "\\pbox[c][][c]{3in}{\\smallskip{}MLCT\\\\Adjusted
                              \\\\No Mosaic}")
  print( xtable( areasCt2 / 10^6,
                caption= "Effect of NLCD offsets on total acreages, $A_{min
                   }=0.5$",
                label= "tab:areas2",
                digits= 1),
        size= "small",
        add.to.row= list(
          pos= list( 0, nrow( areasCt)),
          command= rep("\\noalign{\\smallskip}", times= 2)),
        sanitize.colnames.function= function(x) x)
})
```

```
if( overwriteFigures) areasPlotAdj <-
  qplot( map, acres /10^6,
        data= subset( areasDf2,
          class != "total" & map != "nlcdOffsets"),
        geom="bar", position= "stack",
        fill= class,
        stat="summary", fun.y="sum") +
  scale_fill_manual( "",
                    values= peelLegend[ legendOrder],
                    breaks= names( peelLegend)[ legendOrder]) +
  scale_y_continuous( "M␣acres",
      limits= c(0,2000)) +
  theme_bw( base_family= "serif") +
  scale_x_discrete( "",
      limits= rasterNames2[ rasterNames2 != "nlcdOffsets"],
      breaks= rasterNames2[ rasterNames2 != "nlcdOffsets"],
      labels= expression("Agland2000", "NLCD",
          atop( atop( textstyle( "MLCT"),
                      textstyle( A[ min] ==0.5)),
               phantom(0)),
          atop( atop( textstyle( "MLCT"),
                      textstyle( A[ min] ==0.5)),
               "No␣Mosaic"),
          atop( atop( textstyle( "MLCT"),
                      textstyle( A[ min] ==0.5)),
               "Adjusted"),
          atop( atop( textstyle( "MLCT"),
                      textstyle( A[ min] ==0.5)),
               scriptstyle( "Adjusted,␣No␣Mosaic"))))

cropScatAdjDf <-
```

```
  data.frame( as( stack(getPeelBand( mlctAdj$nomos, "crop"),
                        aglandCrop,
                        raster::mask(acres, cusaMask)),
                 "SpatialGridDataFrame"))
colnames(cropScatAdjDf) <-
  c( "mlctAdj", "agland", "acres", "lon", "lat")
cropScatAdjDf$weight <- with( cropScatAdjDf, acres/ max(acres))




##################################################
### chunk number 21: fig_offsets
##################################################
#line 1113 "/home/nbest/thesis/analysis.Rnw"

if( overwriteFigures) {
  offsetsPlot <-
    qplot( class, acres /10^6,
        data= subset( areasDf2,
          map == "nlcdOffsets" & class != "total"),
        geom= "bar",
        fill= class) +
    scale_fill_manual( "",
        values= peelLegend,
        breaks= names( peelLegend)) +
    scale_y_continuous( "Ma", limits=c( -50, 80)) +
    scale_x_discrete( "", breaks= c( names( peelClasses), "total")) +
    coord_flip() +
    theme_bw( base_family= "serif") +
    opts( legend.position= "none")
  setwd( texWd)
  ggsave( "fig_offsets.pdf",
```

```
          device= pdf,
          plot= offsetsPlot,
          height= 4.5,
          width= 4.5)
}
```

```
################################################
### chunk number 22: fig_areasAdj
################################################
#line 1182 "/home/nbest/thesis/analysis.Rnw"

if( overwriteFigures) {
  my.ggsave( texWd, "fig_areasAdj.pdf",
             device= pdf,
             plot= areasPlotAdj,
             width=  6,
             height= 6)
}
```

```
################################################
### chunk number 23: fig_hexPlotAdj
################################################
#line 1202 "/home/nbest/thesis/analysis.Rnw"

if( overwriteFigures) {
  my.ggsave( texWd, "fig_hexPlotAdj.pdf",
```

```
        device= pdf,
        plot= hexPlot1 %+% cropScatAdjDf +
             aes( agland, mlctAdj) +
             scale_x_continuous( "Agland2000",
                 expand= c( 0,0.0125)) +
             scale_y_continuous( expression( paste( "MLCT␣Adjusted", phantom(
                A[min])))),
                 limits= c( 0, 1),
                 breaks= seq( 0, 1, by= 0.2),
                 expand= c( 0,0.0125)) +
             coord_equal())
}



##################################################
### chunk number 24: fusion
##################################################
#line 1228 "/home/nbest/thesis/analysis.Rnw"

## thumbAgland <- crop( agland,
##                     extent(-83.5, -(82+25/60), 42+55/60, 44+5/60),
##                     filename= "thumbAgland.tif",
##                     progress="text")

nomosCrop <- getPeelBand( mlctAdj$nomos, "crop")
aglandCrop <- unstack( agland)[[ 1]]

## nomosTruth is the sum of the classes from the NLCD offsets.
## other classes adjusted from now on cannot exceed $1 - nomosTruth

if( overwriteRasters) {
```

```
  nomosTruth <- overlay( getPeelBand( mlctAdj$nomos, "water"),
                         getPeelBand( mlctAdj$nomos, "wetland"),
                         getPeelBand( mlctAdj$nomos, "urban"),
                         fun= sum,
                         filename= "nomosTruth.tif",
                         overwrite= TRUE)
} else nomosTruth <-
  raster( list.files( dataPath,
                      patt="nomosTruth.tif",
                      full.names= TRUE))


nomosClasses <- layerNames( mlctAdj$nomos)[ -9]
                                        # leaves out 'total'
                                        # mosaic is already gone

## This is the overlay() bug.  These values of peelCrop should be the same
## but they're not.

## peelCrop <-
##   overlay( aglandCrop, nomosCrop, nomosTruth, fun=
##            function( a, n, t) {
##              ifelse( is.na( a), n, min( a, 1 -t))
##            },
##            filename= "peelCrop.tif",
##            overwrite= TRUE)

peelCrop <-
  if( overwriteRasters) {
    calc( stack( aglandCrop, nomosCrop, nomosTruth), fun=
         function( st) {
           a <- st[ 1]
```

```
          n <- st[ 2]
          t <- st[ 3]
          ifelse( is.na( a), n, min( a, 1 -t))
        },
        filename= "peelCrop.tif",
        overwrite= TRUE)
  } else raster( "peelCrop.tif")


offsetStack <-
  stack( llply( nomosClasses,
              function( class) {
                if( class =="crop")
                  peelCrop
                else
                  zeroes
              }))


noncropFactor <-
  overlay( peelCrop, nomosCrop, nomosTruth, fun=
         function( p, n, t) {
           ifelse( 1 -n -t <= 0,
                   0,
                   ( 1 -p -t) /( 1 -n -t))
         })


factorStack <-
  stack( llply( nomosClasses,
              function( class) {
                if( class == "crop")
```

```
                  zeroes
                else if( class %in%
                        c( "water", "wetland", "urban"))
                  ones
                else
                  noncropFactor
              }))

## stupid overlay bug!
##
## aglandComplete <-
##   if( overwriteRasters || TRUE) {
##     overlay( stack( unstack( mlctAdj$nomos)[ -9]),
##              factorStack,
##              offsetStack,
##              fun= function( x, m, b) m *x +b,
##              filename= "aglandComplete.tif",
##              overwrite= TRUE,
##              progress= "text")
##   } else brick( list.files( rasterWd,
##                             patt="^aglandComplete.tif$",
##                             full.names=TRUE))
## layerNames( aglandComplete) <- names(peelClasses)[-8]

aglandComplete <-
  stack( unstack( mlctAdj$nomos)[ -9]) *factorStack +offsetStack
aglandComplete <- writeRaster( aglandComplete,
                               "aglandComplete.tif",
                               overwrite= TRUE)
layerNames( aglandComplete) <- names(peelClasses)[-8]

aglandCompleteSum <- sum( aglandComplete)
```

```
## lt1AgcTotal <- extract( stack( peelCrop,
##                                nomosCrop,
##                                nomosTruth,
##                                noncropFactor,
##                                aglandComplete,
##                                aglandCompleteSum),
##                         which( aglandCompleteSum[] < 0.999))

## colnames( lt1AgcTotal)[ 1:4] <- c( "p", "n", "t", "factor")
## lt1AgcTotal <- data.frame( lt1AgcTotal)
## lt1AgcTotal <- within( lt1AgcTotal, { term1 <- n-p; term2 <- 1-n-t})

## within( head( lt1AgcShrub), {


## normalize to fix 65 pixels missing area
aglandComplete <- aglandComplete /aglandCompleteSum
layerNames( aglandComplete) <- names(peelClasses)[-8]

agcMap <- coverMaps( aglandComplete, 0.4,
                    classes= layerNames( aglandComplete)[1:4]) +
  coord_equal() +
  facet_grid( variable ~ .)

agcMap2 <- coverMaps( aglandComplete, 0.4,
                    classes= layerNames( aglandComplete)[5:8]) +
  coord_equal() +
  facet_grid( variable ~ .)
```

```
##################################################
### chunk number 25: fig_agc
##################################################
#line 1399 "/home/nbest/thesis/analysis.Rnw"

my.ggsave( texWd, "fig_agc.png",
          plot= agcMap, width=4.5, height=8)


##################################################
### chunk number 26: fig_agc2
##################################################
#line 1413 "/home/nbest/thesis/analysis.Rnw"

my.ggsave( texWd, "fig_agc2.png",
          plot= agcMap2, width=4.5, height=8)


##################################################
### chunk number 27: table_rmse3
##################################################
#line 1429 "/home/nbest/thesis/analysis.Rnw"

setwd( rasterWd)

rmseDf3 <-
  cbind( agland=c( TRUE, rep(FALSE, times=3)),
        rbind( c( TRUE, 0.5,
                  rmseRast( getPeelBand( aglandComplete, "crop"),
                            aglandCrop)),
               rmseDf2))
```

```
rmseDf3 <- within(rmseDf3, offset <- as.logical( offset))
                                        # had to change offset column back
                                        # to true/false;  maybe this can be
                                        # avoided with list() instead of c()

rmseXt <- xtable( rmseDf3,
                 caption= "RMSE␣of␣PEEL␣vs.␣Agland2000",
                 label= "tab:rmse3",
                 digits= c( 0, 0, 0, 1, 3))
                                        # looks like some kind of bug in
                                           xtable()
                                        # manual correction:
rmseXt$agland <- rmseDf3$agland
rmseXt$offset <- rmseDf3$offset

print( rmseXt,
      include.rownames= FALSE,
      sanitize.colnames.function= function(x) x)



##################################################
### chunk number 28: tab_areas3
##################################################
#line 1459 "/home/nbest/thesis/analysis.Rnw"

areasCt3 <- acreageTable( c( rasterNames2[ c( 1, 2, 4, 7)], "aglandComplete"))

local({
  colnames( areasCt3) <-
    c( "Agland2000", "NLCD",
      "\\pbox[c][][c]{3in}{MLCT\\\\No␣Mosaic}",
```

```
      "\\pbox[c][][c]{3in}{\\smallskip{}MLCT\\\\Adjusted\\\\No␣Mosaic}",
      "PEEL")
  print( xtable( areasCt3 / 10^6,
                 caption= "PEEL␣acreages,␣$A_{min}=0.5$",
                 label= "tab:areas3",
                 digits= 1),
         size= "small",
         add.to.row= list(
           pos= list( 0, nrow( areasCt)),
           command= rep("\\noalign{\\smallskip}", times= 2)),
         sanitize.colnames.function= function(x) x)
  ##,
  ##      floating= FALSE)
})

cropScatAgcDf <-
  data.frame( as( stack(getPeelBand( aglandComplete, "crop"),
                        aglandCrop,
                        raster::mask(acres, cusaMask)),
                 "SpatialGridDataFrame"))
colnames(cropScatAgcDf) <-
  c( "agc", "agland", "acres", "lon", "lat")
cropScatAgcDf$weight <- with( cropScatAgcDf, acres/ max(acres))



cropScatAgcDf <-
  within( cropScatAgcDf,
          { cat <- NA
            cat[    agc == 0] <- 0
            cat[ agc > 0 & is.na( agland)] <- 1
            cat[ agc > 0 & ( agc -agland) < 0.1] <- 2
```

```
          cat[ agc > 0 & agc  < agland] <- 3
          cat[ agland == 1] <- 4
        })

agcThemeMap <-
  ggplot( cropScatAgcDf[ !is.na( cropScatAgcDf$cat),],
        aes( x= lon, y= lat)) +
  geom_tile( aes( fill= factor( cat))) +
  scale_fill_brewer( "",
                   breaks= 0:4,
                   labels= c(
                     "PEEL = 0",
                     "PEEL > 0, Ag2k is null",
                     "PEEL > 0, PEEL = Ag2k",
                     "PEEL > 0, PEEL < Ag2k",
                     "Ag2k = 1")) +
  coord_equal() +
  theme_map

##################################################
### chunk number 29: fig_hexPlotAgc
##################################################
#line 1527 "/home/nbest/thesis/analysis.Rnw"

if( overwriteFigures) {
  setwd( texWd)
  ggsave( "fig_hexPlotAgc.pdf",
        device= pdf,
        plot= hexPlot1 %+% cropScatAgcDf +
```

```
            aes( agland, agc) +
          scale_fill_gradientn( colours= brewer.pal( 6, "YlGn"),
                    trans= "log10",
                    limits=c( 10, 20000)) +
            scale_x_continuous( "Agland2000") +
            scale_y_continuous( "PEEL",
                limits= c( 0, 1),
                breaks= seq( 0, 1, by= 0.2)) +
            coord_equal(),
         height= 4.5,
         width= 4.5)
}



##################################################
### chunk number 30: fig_agcThemeMap
##################################################
#line 1558 "/home/nbest/thesis/analysis.Rnw"

if( overwriteFigures) {
  my.ggsave( texWd, "fig_agcThemeMap.png", width=7.5,
            plot= agcThemeMap,
            bg= "transparent")
}



##################################################
### chunk number 31: crop_cats
##################################################
#line 1602 "/home/nbest/thesis/analysis.Rnw"
```

```
setwd( rasterWd)

cropCats <-
  c("cereals", "field_crop", "forage", "maize",
    "rice", "shrub_crop", "soybean", "sugarcane",
    "tree_crop", "wheat")
names( cropCats) <- cropCats


cropCatsPeel <-
  list( crop= c( "cereals", "field_crop", "forage",
                 "maize", "rice", "soybean",
                 "sugarcane", "wheat"),
        open= NULL,
        shrub= "shrub_crop",
        forest= "tree_crop")

cropCats <- llply(  cropCats, function(c) {
  raster( paste( dataPath,
                 paste( c, "tif", sep="."),
                 sep= "/"))
})

cropStack <- stack( cropCats)


cropSum <- overlay( stack( cropCats[ cropCatsPeel$crop]),
                    fun= sum)

cropSum[ is.na( cropSum[])
         & !is.na( getPeelBand( aglandComplete,
```

```
                                "crop")[])
        ] <- 0
cropSum <- writeRaster( cropSum,
                    file= "cropSum.tif",
                    overwrite= TRUE)

cropNormalFunc <- function( st) {
  cropCat <- st[ 1]
  cropSum <- st[ 2]
  agc <-     st[ 3]
  ifelse( cropSum == 0,
          0,
          cropCat / cropSum)
}

cropNormal <- stack( llply( cropCats[ cropCatsPeel$crop],
                            function( crop) {
                              calc( stack( crop,
                                           cropSum,
                                           getPeelBand( aglandComplete, "crop"))
                                           ,
                                    fun= cropNormalFunc)
                            }))

cropSubClasses <- getPeelBand( aglandComplete, "crop") *cropNormal
layerNames( cropSubClasses) <- cropCatsPeel$crop




################################################
### chunk number 32: fig_cropSubClassesMap
```

```
##################################################
#line 1669 "/home/nbest/thesis/analysis.Rnw"

if( overwriteFigures) {
  cropSubClassesMap <-
    coverMaps( cropSubClasses, 0.4,
              classes= layerNames(cropSubClasses)[ 1:4]) +
    coord_equal() +
    facet_grid( variable ~ .)
  my.ggsave( texWd, "fig_cropSubClassesMap.png",
            plot= cropSubClassesMap)
}



##################################################
### chunk number 33: fig_cropSubClassesMap2
##################################################
#line 1690 "/home/nbest/thesis/analysis.Rnw"

if( overwriteFigures) {
  cropSubClassesMap2 <-
    coverMaps( cropSubClasses, 0.4,
              classes= layerNames(cropSubClasses)[ 5:7]) +
    coord_equal() +
    facet_grid( variable ~ .)
  my.ggsave( texWd, "fig_cropSubClassesMap2.png",
            plot= cropSubClassesMap2)
}
```

```
##################################################
### chunk number 34: restack_crops
##################################################
#line 1733 "/home/nbest/thesis/analysis.Rnw"


check <-
  getPeelBand( aglandComplete, "crop") -
  sum( cropSubClasses)

mlctCrop <-
  overlay( getPeelBand( aglandComplete, "crop"),
           cropSum,
         fun= function( agc, ag) {
           ifelse( is.na( ag), agc, 0)
         })

check <-
  getPeelBand( aglandComplete, "crop") -
  sum( cropSubClasses) -
  mlctCrop


peelData <- stack( aglandComplete, cropSubClasses, mlctCrop, check)
layerNames( peelData)[ 16:17] <- c("other_crop", "check")

peelDf <- data.frame(peelData[])

noisyCells <- rownames( with( peelDf,
    peelDf[ !is.na( check) & check >= 0.001,]))
# 325 cells with noise above this threshold
```

```
mlctCrop <- mlctCrop + check

peelData <- stack( aglandComplete, cropSubClasses, mlctCrop)
layerNames( peelData)[ 16] <- c("mlct_crop")

## peelData <- writeRaster( peelData,
##                          filename= sprintf( "%s/peel.tif", rasterWd),
##                          overwrite= TRUE)

peelData <- brick( peelData,
                  filename= sprintf( "%s/peel.tif", rasterWd),
                  overwrite= TRUE)

layerNames( peelData) <-
    c( unlist( llply( c( aglandComplete,
                         cropSubClasses),
                      layerNames)),
      "mlct_crop")


peelDf <- data.frame( peelData[ !is.na( grid[])])
rownames( peelDf) <- grid[][ !is.na(grid[])]

write.csv( format.df( peelDf,
                      dec=3,
                      numeric.dollar=FALSE,
                      na.blank= TRUE),
          file= "peel.csv",
          quote= FALSE)

## copy to data archive
## file.copy( "peel.csv", "~/see/data/cimdb/peel_thesis.csv",
```

```
##           overwrite=TRUE)


## peelData <- brick( aglandComplete, cropSubClasses)



###################################################
### chunk number 35: cleanup
###################################################
#line 1802 "/home/nbest/thesis/analysis.Rnw"
options( prompt= ">␣", continue= "+␣", width= 80)
```