Imperial College London
Department of Civil and Environmental Engineering

Toward integrated modelling systems to assess vulnerability of water resources under environmental change

Simon Moulds

August 2016

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Civil and Environmental Engineering of Imperial College London
and the Diploma of Imperial College London

Copyright declaration

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.

Abstract

Land use, land cover and land management change threatens the sustainability of ecosystem services upon which individuals and communities depend. However, quantifying the effects of large-scale environmental change on regional water resources and climate is challenging because of a lack of appropriate data as well as fundamental limitations of environmental models. This thesis focuses on the development of integrated modelling systems for representing feedback mechanisms between human activities and the environment at regional scales. India is selected as a case study because of the unprecedented scale of environmental change in this country over recent decades.

Land use change modelling is identified as a viable method for reconstructing historical land use/land cover at regional scales. This is facilitated through the development of a new modelling framework which allows users to perform the entire modelling workflow in the same environment and provides a consistent interface to different spatial allocation models. Hence, the modelling framework enables model intercomparison and ensemble experiments. It furthermore promotes reproducible science because it allows applications to be expressed programmatically.

An adapted version of the Change in Land Use and its Effects (CLUE) land use change model is used to reconstruct historical land use/land cover in India between 1956–2010. The model algorithm explicitly accounts for competition between land use/land cover categories as a result of dynamic socio-economic and biophysical conditions. A further dataset showing the irrigated area of various crops is developed by spatially disaggregating historical agricultural inventory data based on maps of cropland extent and biophysical suitability. Land use/land cover maps are supplied to an offline historical simulation of the Joint UK Land and Environment Simulator (JULES), a process-based land surface model, to generate soil moisture values across the Gangetic plain. Simulated soil moisture values are modified to account for the effects of irrigation. The procedure exploits the characteristics of the irrigated area dataset in order to account for the growing season of individual crops.

Existing tools for coordinating complex workflows in the hydrological sciences are strongly coupled to underlying modelling frameworks. As a result, they lack flexibility and often necessitate refactoring of the source code of model components. Exploring these issues further, an experiment is devised in which the data processing language R is set up as a workflow orchestration tool for hydrological data analysis and modelling. A new software package implements a set of classes for representing multi-dimensional hydrological data and to provide a common interface to hydrological models. The experimental set-up is demonstrated through two example applications drawn from hydrology and the emerging discipline of socio-hydrology. These serve to highlight the flexibility of the R system for workflow orchestration and model coupling but also draw attention to several areas for future development.

Statement of own work

The work presented in this submission is my own. Any use of the work of others, whether published or unpublished, is duly acknowledged by reference to the sources.

The following extracts of this thesis have been published or are in the latter stages of preparation for submission to peer-reviewed journals:

Chapter 2

Moulds, S., Buytaert, W. and Mijic, A. (2015). An open and extensible framework for spatially explicit land use change modelling: the lulcc R package. *Geoscientific Model Development*, 8:3215-3229, doi:10.5194/gmd-8-3215-2015.

Chapter 3

Moulds, S., Buytaert, W. and Mijic, A. (2016). A spatio-temporal land use/land cover reconstruction for India (1956 – 2010). *Scientific Data*. In preparation.

Chapter 4

Moulds, S., Buytaert, W. and Mijic, A. (2016). Mapping India's green revolution: a spatio-temporal dataset of irrigated land, 1956 – 2010. *Hydrology and Earth System Sciences*. In preparation.

Acknowledgements

Funding for this research project was gratefully received from the Grantham Institute for Climate Change and the Environment. Additional support, notably towards an extended research visit to Ecuador, was provided by Climate-KIC. My PhD research has contributed to Hydroflux India, a project which forms part of the NERC Changing Water Cycle programme (NE/I022558/1).

I am grateful to my supervisors Dr Wouter Buytaert and Dr Ana Mijic for sharing with me their expertise, experience and enthusiasm over the last few years. Their friendship and encouragement have made this PhD a tremendously enjoyable experience. I am delighted that we will continue our collaboration, albeit from afar, in the coming months and years.

It has been a huge privelege to be a part of Hydroflux India and I extend my thanks to the wider team for many successful collaborations over the course of the project. I would particularly like to thank Prof. Rajiv Sinha of IIT Kanpur for organising an informative and immensely enjoyable field visit to the Ganga basin at the start of this project, as well as Prof. Pradeep Mujumdar for his warm hospitality on our many visits to IISc Bangalore.

I thank my colleagues past and present in the Environmental and Water Resource Engineering section for their camaraderie and good humour. I would particularly like to mention Jimmy O'Keeffe, Jonny Ritson, Bastian Manz and Gina Tsarouchi for their companionship on various travels.

My deepest thanks are to my wife Emma for her constant and unconditional love and support, without which none of this would have been possible. I would also like to thank my parents for the sacrifices they have made to provide me with the opportunities which have brought me here.

Table of Contents

Copyright declaration **2**

Abstract **3**

Statement of own work **4**

Acknowledgements **5**

Table of Contents **6**

List of Figures **9**

List of Tables **12**

1 Introduction **13**

1.1 Environmental change in India 13

1.1.1 The green revolution 13

1.1.2 The South Asian monsoon 16

1.2 Hydroflux India 18

1.3 Aim and objectives 20

1.3.1 Aim 20

1.3.2 Objectives 20

1.4 Thesis outline 21

2 Land use change modelling **23**

2.1 Literature review 24

2.2 Software description 26

2.2.1 Data 31

2.2.2 Data processing 31

2.2.3 Predictive modelling 36

2.2.4 Demand 41

2.2.5 Allocation 43

2.2.6 Pattern validation 48

2.3 Discussion 49

3 Modelling historical land use/land cover change in India 55
3.1 Literature review . . . 56
3.2 Methodology . . . 58
3.2.1 Input data . . . 58
3.2.2 Data preparation . . . 61
3.2.3 Land use change modelling procedure . . . 64
3.3 Results . . . 66
3.3.1 Statistical analysis . . . 66
3.3.2 Spatial allocation . . . 66
3.4 Discussion . . . 67
3.5 Chapter summary . . . 74

4 Reconstructing historical changes in India's irrigated area 76
4.1 Literature review . . . 77
4.2 Methodology . . . 79
4.2.1 Input data . . . 79
4.2.2 Data preparation . . . 83
4.2.3 Spatial disaggregation . . . 83
4.3 Results . . . 87
4.4 Discussion . . . 87
4.5 Chapter summary . . . 94

5 Quantifying the impact of land change on the South Asian monsoon 95
5.1 Literature review . . . 96
5.1.1 Surface energy and water balance . . . 96
5.1.2 Role of vegetation in soil moisture feedbacks . . . 97
5.1.3 Soil moisture-climate interactions and feedbacks . . . 98
5.1.4 Hydrometeorological feedbacks in the Indo-Gangetic plains . . . 98
5.2 Methodology . . . 105
5.2.1 Joint-UK Land and Environment Simulator . . . 105
5.2.2 Input data . . . 106
5.2.3 Land surface modelling . . . 107
5.2.4 Post-processing . . . 108
5.3 Results . . . 111
5.4 Discussion . . . 111

6 Hydrological model coupling and workflow orchestration in R 122
6.1 Literature review . . . 123
6.1.1 Current trends in hydrological modelling . . . 123
6.1.2 Modelling complex systems . . . 127

6.2 Software description . . . 131
6.2.1 Classes . . . 131
6.2.2 Methods . . . 135
6.3 Example applications . . . 142
6.3.1 Plynlimon . . . 142
6.3.2 Reservoir operations . . . 148
6.4 Discussion . . . 157

7 Summary and conclusions **162**
7.1 Summary of thesis . . . 162
7.1.1 Modelling land use/land cover change . . . 162
7.1.2 Mapping irrigated land . . . 163
7.1.3 Land surface modelling of the Ganga basin . . . 164
7.1.4 Hydrological modelling and workflow orchestration . . . 165
7.2 Summary of contributions to knowledge . . . 166
7.3 Recommendations for further work . . . 166
7.3.1 Recommendations concerning the *lulcc* modelling framework . . . 166
7.3.2 Recommendations concerning future projections of land change . . . 167
7.3.3 Recommendations concerning improvements to irrigated area mapping 168
7.3.4 Recommendations concerning the inclusion of anthropogenic activities in Earth system modelling . . . 168
7.3.5 Recommendations concerning the development of the R system for workflow orchestration . . . 169
7.4 Final remarks . . . 169

References **171**

Appendices **212**

A Manual for *lulcc* **213**

B Manual for *Hydro* **274**

List of Figures

1.1 Change in net cropped area, net irrigated area and gross irrigated area in India between 1951–2012 15
1.2 Hydroflux India model setup 20

2.1 Flowchart showing general methodology for inductive land use change modelling 28
2.2 Class diagram showing the relationship between *raster* and *lulcc* classes. . . 29
2.3 Class diagram for the *lulcc* package. 30
2.4 Observed land use/land cover maps for the Plum Island Ecosystems site in 1985, 1991 and 1999 34
2.5 Receiver Operator Characteristic curves for predictive models of forest, built and other land uses 39
2.6 Receiver Operator Characteristic curve for predictive model for the gain in build land 41
2.7 Suitability of pixels in the Plum Island Ecosystems study site to belong to Forest, Built and Other land use classes 42
2.8 Agreement budget for the transition from Forest to Built for CLUE-S and ordered model outputs considering reference maps at 1985 and 1999 and simulated map for 1999. 50
2.9 Figure of merit scores corresponding to the agreement budgets depicted in Figure 2.8. 51

3.1 Survey methods employed to collect district-level land use/land cover and agricultural inventory data across India 60
3.2 Comparison of the area of crop, forest, grass and barren/sparsely vegetated land use/land cover classes at the national level considering the raw inventory data, inventory data after quality control and calibration with GlobeLand30 product, and area actually allocated. 68
3.3 Comparison of the area of urban land use at the national level considering the inventory data after calibration with GlobeLand30 product and the area actually allocated. 69

3.4 Land use/land cover in 2010, derived from the GlobeLand30 dataset (Chen et al., 2015). 70
3.5 Location of change of crop, forest, grass and barren/sparsely vegetated land use/land cover classes between 1960–2010. 71
3.6 Spatial distribution of fallow land between 1999–2004, highlighting the impact of the July 2002 drought on cropland extent, especially in parts of north-west India. 72

4.1 Mean Indiastat data availability in each district unit over 5 year time slices. 81
4.2 Suitability maps for high-input irrigated wheat and rice from FAO GAEZ database. 82
4.3 Input data to disaggregation method for Allahabad district: **a** Fractional membership of each pixel to the district; **b** FAO GAEZ suitability of each pixel to irrigated, high-input wheat; **c** Fraction of each pixel devoted to cropland in the year 2000; **d** Change in area of irrigated wheat over the course of the study period. 85
4.4 Change in irrigated area of various crops in India at the national level between 1950-51 and 2010-11. 88
4.5 Irrigated wheat area at various time points during the study period. 89
4.6 Irrigated rice area at various time points during the study period. 90
4.7 Irrigated sugarcane area at various time points during the study period. . . 91
4.8 Comparison between maps of irrigated wheat, rice and sugarcane from the current study and corresponding maps from the MIRCA2000 dataset. 92

5.1 Conceptual diagram of different evapotranspiration regimes 99
5.2 Study region for land surface modelling. 108
5.3 Land cover in the Ganga basin for the year 2000. 110
5.4 Land cover change in the Ganga basin, 1971–2004. 112
5.5 Change in net irrigated area in the Ganga basin, 1971–2004. 113
5.6 Net irrigated area in the Ganga basin at 5 year intervals, 1971–2004. 114
5.7 Monthly irrigated area in the Ganga basin, June 2000–May 2001. 115
5.8 Mean monthly difference between soil moisture values simulated by JULES and the corresponding values adjusted to include irrigation, June 2000–May 2001. 116
5.9 Mean seasonal difference between soil moisture values simulated by JULES and the corresponding values adjusted for irrigation, at various time points during the study period. 117
5.10 Total volume of water in the top 0.1m soil layer across the Ganga basin between 1999–2005. 118

6.1 Class diagram showing the relationship between *spacetime* and *Hydro* classes. 133
6.2 Class diagram for the *Hydro* package. 134
6.3 Mean daily reference evapotranspiration during the first four months of 2008 across four weather stations in the Plynlimon catchments in Wales, United Kingdom. 149
6.4 Results of the reservoir operations model simulation. 156

List of Tables

2.1 Functions included in the *lulcc* package 32

3.1 Translation of land use/land cover classification schemes of inventory data and GlobeLand30 product to the classes used in the CLUE modelling procedure. 63
3.2 Performance of random forest models of the spatial distribution of forest, urban, agricultural uses, grassland and barren/sparsely vegetated land in India 66

4.1 Crop types included in the present study and the corresponding GAEZ suitability map. 84

5.1 Literature review of methods for representing the effects of irrigation on soil moisture at regional scales. 103
5.2 Translation of land use/land cover classification scheme to JULES land cover classes. 109

6.1 Methods included in the Hydro package 136
6.2 Names of model components of reservoir operations model of Garcia et al. (2016) with reference to the corresponding model equation. 151

1 Introduction

Land use, land cover and land management change, collectively referred to as "land change" in the remainder of this thesis, is cumulatively an important driver of global and regional environmental change (Lambin and Meyfroidt, 2011). Land use refers to the way in which humans exploit land resources, land cover describes the physical material at the land surface and land management encompasses the various practices that are used to ensure land resources meet human requirements. The conversion of rainfed agriculture and natural land cover to intensively managed agricultural systems across much of South Asia has put regional water resources under severe pressure (Rodell et al., 2009; Shankar et al., 2011; Wada et al., 2012). Land change may also influence local and regional climate through direct and indirect impacts on the surface energy and water balance (Pitman et al., 2009; Seneviratne et al., 2010; Boysen et al., 2014). However, attempts to quantify the impact of land change on regional water resources and climate are limited by a lack of data showing the rate and location of change as well as fundamental limitations of current environmental models. This thesis addresses these deficiencies by developing integrated modelling systems for representing interactions and feedbacks between anthropogenic activities and the environment at the regional level. India is selected as a case study because of the unprecedented scale of land change here over recent decades as well as the range of issues that are present. The remainder of this introductory chapter describes the case study and places the research in the context of Hydroflux India; a large and interdisciplinary project to which this thesis contributes. It presents the aim and objectives that will guide the research and lastly summarises the thesis structure.

1.1 Environmental change in India

1.1.1 The green revolution

India's green revolution, initiated in the mid-1960s to achieve food security for its growing population, was characterised by the introduction of high-yielding wheat and rice varieties combined with increasing use of fertilisers, pest-control and agricultural machinery (Evenson and Gollin, 2003; Alauddin and Quiggin, 2008; Pingali, 2012). Irrigation, which provides a buffer against intraseasonal monsoon variability and allows farmers to grow crops

in the dry winter season, has been central to the resulting productivity gains (Thenkabail et al., 2009). Figure 1.1 compares the change in net cropped area (the area used to grow crops at least once during a given year) with net irrigated area (the area irrigated at least once during a given year) and gross irrigated area (the total irrigated area under various crops, counting irrigated area as many times as the number of irrigated crops grown during the same year) between 1956–2012. It shows that, while agricultural expansion was important in the decade prior to the green revolution, the most substantial environmental change thereafter was the intensification of agricultural systems. The epicentre of the green revolution was the fertile Indo-Gangetic plains in northern India, particularly the states of Punjab, Haryana and western parts of Uttar Pradesh (Sarkar et al., 2011). The green revolution has enabled India to become self-sufficient in food production while supporting the livelihoods of millions and reducing the incidence of rural poverty (Alauddin and Quiggin, 2008; Sarkar et al., 2011; Varua et al., 2016). Today, more than 60% of the working population are employed in the agricultural sector (Singh et al., 2014), which contributes around 18% to the gross domestic product of India and supplies more than 70% of its exports (Krishna Kumar et al., 2004; Prasanna, 2014; Amrith, 2016).

However, besides the manifest benefits, the green revolution has induced large-scale environmental change which now threatens the sustainability of ecosystem services vital to India's food and water security (Sarkar et al., 2011). In the first half of the twentieth century and immediately following independence in 1947 there was considerable public expenditure on large hydraulic structures including dams and canals for surface water irrigation schemes (Amrith, 2016). The supply-driven canal system was, however, unable to meet the irrigation requirements of farmers growing high-yielding rice and wheat varieties (Humphreys et al., 2010). The availability of small, affordable pumps and boring rigs from around 1970 proved a technological breakthrough which enabled groundwater to emerge as the primary source of water for irrigation (Scott and Sharma, 2009; Amrith, 2016). Groundwater provides on-demand irrigation throughout the year and is more resilient to drought conditions compared with surface water irrigation schemes (Shah, 2009; Amrith, 2016). The number of electric or diesel pumpsets in operation increased from around 150,000 in 1950 to 19 million in 2000 (Shah, 2009), largely due to a policy of rural electrification initiated by State governments around the time of the green revolution (Scott and Sharma, 2009; Amrith, 2016). According to government estimates, groundwater is used to irrigate more than 60% of the gross cropped area in India (Scott and Sharma, 2009; Shankar et al., 2011). In north-west India, Shah et al. (2006) found that more than 90% of the cultivated land is irrigated, of which around 90% is supplied from groundwater. Satellite data from the Gravity Recovery and Climate Experiment (GRACE) reveal a gradual depletion of groundwater resources in north-west India, especially Haryana and western Uttar Pradesh (Rodell et al., 2009; Panda and Wahr, 2016; Long et al., 2016), while

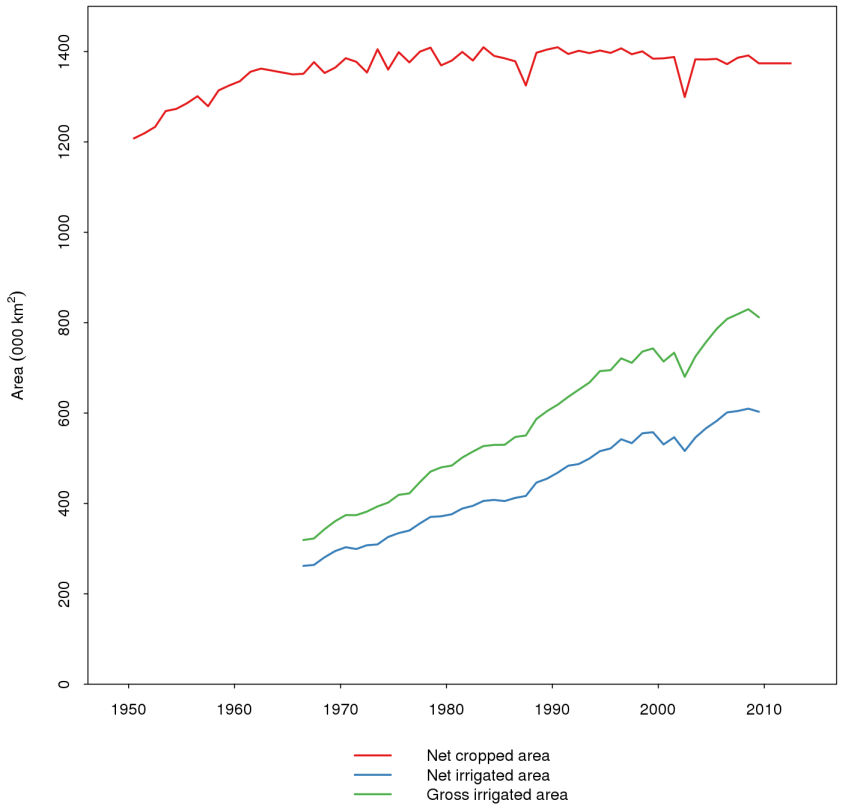


Figure 1.1.: Change in net cropped area, net irrigated area and gross irrigated area in India between 1951–2012. Data obtained from Indiastat (Indiastat, 2016) and the ICRISAT Village Dynamics in South Asia (VDSA) database (Icrisat, 2016). Values for net irrigated area and gross irrigated area are only available for the period 1966–2009

.

various modelling studies show unsustainable groundwater use across the Indo-Gangetic plains and the Indian peninsula (Wada et al., 2012; Döll et al., 2014). A recent analysis of observed groundwater level data, while broadly supporting these conclusions, highlights the spatial heterogeneity in groundwater depletion across the basin (MacDonald et al., 2016). Groundwater depletion threatens the livelihoods of small and marginal farmers because installing, maintaining and operating wells becomes more expensive as the water table falls (Hira, 2009). It can also lead to declining groundwater quality either because of upwelling of saline groundwater or because changing groundwater dynamics causes saline groundwater to flow into fresh groundwater (Scott and Sharma, 2009; Humphreys et al., 2010).

The pressure on water resources in India will intensify in the near future. According

to the United Nations Population Division, population growth in India is not expected to stabilise until around 2060, when its population will have grown from 1.21 billion following the 2011 census to 1.72 billion, with the highest growth rate occurring in urban areas (James, 2011; Gururaja and Sudhira, 2012). Continued economic growth is likely to increase industrial water use (Hanjra and Qureshi, 2010; Pingali, 2012). At the same time, land use/land cover change arising from urbanisation and industrialisation is reducing the area under agriculture in many parts of the country (Alauddin and Quiggin, 2008; Gururaja and Sudhira, 2012). Groundwater extraction for domestic usage in urban areas will put additional strain on water resources (Alauddin and Quiggin, 2008). Rising demand for biofuels, in part due to policies enacted by the Indian government (Ravindranath et al., 2011), is putting pressure on the agricultural system and may decrease the area of land available for food production (Fraiture et al., 2008; Naylor, 2011). Soil degradation arising from unsustainable management practices has substantially reduced the productivity of large tracts of agricultural land (Singh, 2000; Srivastava et al., 2016).

1.1.2 The South Asian monsoon

The South Asian monsoon, which supplies up to 80% of India's total annual rainfall between June–September (Basu, 2007), is critically important to the nation's water resources (Singh, 2016). The monsoon develops during boreal spring as enhanced solar heating warms the land mass of South and South-East Asia faster than the surrounding oceans and establishes a large-scale meridional temperature gradient (Turner and Annamalai, 2012; Singh et al., 2014). This leads to the formation of a low pressure trough over the Indo-Gangetic plain in late spring which, in turn, drives moisture-laden winds from the Indian Ocean to the Indian subcontinent (Turner and Annamalai, 2012). As warm, moist air rises and cools the condensation of water vapour releases latent heat to the atmosphere, driving further uplift and initiating deep convection throughout the troposphere (Kitoh et al., 2013). Changes in the timing, intensity and duration of the monsoon rainfall as a result of climate change and other factors are a major threat to India's food and water security (Amrith, 2016). The land ocean-temperature contrast, which has been shown to correspond with the strength of the mean monsoon (Li and Yanai, 1996; Chou, 2003), is expected to increase in response to greenhouse gas forcing (Sutton et al., 2007). At the same time the warming of the Indo-Pacific oceans, which has already been observed in recent decades (Knutson et al., 2006), will supply greater amounts of water vapour to the monsoon region (Turner and Annamalai, 2012). These factors would tend to suggest an increase in monsoon rainfall (Turner and Annamalai, 2012). However, since the middle of the twentieth century there has been a decreasing trend in mean monsoon rainfall over India (e.g. Ramanathan et al., 2005; Bollasina et al., 2011; Turner and Annamalai, 2012). Recently, Krishnan et al. (2015) showed that forcing a global climate model with historical greenhouse gas emis-

sions, aerosol emissions and land use/land cover change improved the ability of the model to simulate observed trends of mean monsoon rainfall compared with simulations that only incorporated greenhouse gas emissions. In addition, several researchers have identified the cooling effect of large-scale irrigation across the Indo-Gangetic plains as a possible mechanism weakening the strength of the monsoon (e.g. Niyogi et al., 2010).

The monsoon is associated with intraseasonal variations between active and break periods of precipitation (Gadgil, 2003; Krishnamurthy and Shukla, 2007), which vary in length from several days to several weeks (Befort et al., 2016). Active monsoon periods are associated with frequent storms which produce heavy, well-distributed rainfall over the monsoon region (Gadgil, 2003), while break periods are associated with the migration of the low pressure trough to the foothills of the Himalayas and the prolonged absence of rainfall (Gadgil, 2003). Analysis of observed rainfall data has shown an increase in the frequency and magnitude of extreme rainfall events since 1951, with a decrease in the frequency of moderate rainfall events (Goswami et al., 2006; Rajeevan et al., 2008; Singh et al., 2014). However, there is considerable spatial heterogeneity in historical trends of extreme rainfall events (Ghosh et al., 2011), with evidence of the strongest positive trends in northern India (Dash et al., 2007). Vinnarasi and Dhanya (2016) identified changes in the seasonal monsoon pattern over the last 50 years, showing the period of maximum frequency of active periods has shifted towards the end of the monsoon season, while break periods are moving towards the start. Meanwhile, Niranjan Kumar et al. (2013), examining the Standardized Precipitation Evapotranspiration Index (SPEI) for India during 1901–2010, identified a significant ($p < 0.05$) increase in the intensity and spatial extent of moderate droughts during the monsoon since 1951, which they attribute to the observed warming of the Indian Ocean. Similarly, Mallya et al. (2016) identified a robust trend of increasing drought severity and frequency over the Indian monsoon region during the period 1972–2004, using the SPEI as well as three other drought indices to analyse two gridded precipitation datasets. These observations are consistent with the general intensification of the hydrological cycle in the tropics (Trenberth et al., 2003).

Variations in the frequency and duration of break periods during the monsoon season can have severe consequences for agricultural production and livelihoods (Annamalai et al., 2007; Turner and Annamalai, 2012; Parker et al., 2016), particularly if they coincide with important stages in the crop growth cycle (Singh et al., 2014). Furthermore, long break periods are associated with decreased mean monsoon rainfall across India (Gadgil, 2003; Goswami et al., 2006), with repercussions for regional water resources (Prasanna, 2014; Singh et al., 2014; Singh, 2016). For example, an extended dry spell around July 2002, in which rainfall was 56% below normal, contributed to a decrease in the mean annual rainfall of 21% (Bhat, 2006), resulting in drought conditions across India and falls in agri-

cultural production and economic growth. With similar events expected to become more frequent under climate change (e.g. Turner and Annamalai, 2012), groundwater will assume a greater importance for protecting food production against intraseasonal monsoon variability. However, at the same time, increasing reliance on groundwater resources will intensify pressure on regional water resources. Thus, as Shah (2009) points out, climate change will increase the dependence of the agricultural sector and other users on groundwater while, at the same time, threatening the sustainability of the resource.

1.2 Hydroflux India

Accurate forecasts about the behaviour of the South Asian monsoon under climate change, especially concerning intraseasonal variability, are critical to enable policy makers to make effective water management decisions (Hasson et al., 2013; Suhas et al., 2013; Kumar, 2015). According to Turner and Annamalai (2012), the variability of the South Asian monsoon at multi-decadal, interannual and intraseasonal time scales "cannot be correctly simulated without accurate representation of the mean state". Thus, it is concerning that simulations of monsoon rainfall of four Coupled Model Intercomparison Project phase 3 (CMIP3) general circulation models (GCMs), chosen for their apparent skill in reproducing key features of the South Asian monsoon, show considerable differences in the mean monsoon rainfall and interannual variability compared with observations (Turner and Annamalai, 2012). Accordingly, there is also large disagreement between models about the behaviour of the monsoon at intraseasonal time scales, particularly the occurrence of protracted break periods and extreme precipitation events (Turner and Annamalai, 2012; Sharmila et al., 2015). Although in many respects historical CMIP5 GCM simulations show marginal improvements compared with those of CMIP3 they are still associated with large uncertainties, particularly regarding intraseasonal variability and the timing of the monsoon onset (Sperber et al., 2013; Saha et al., 2014). The projection of future mean monsoon rainfall from the ensemble-mean of CMIP5 models shows a weak positive trend as a result of the enhanced moisture content of the atmosphere above the Indian Ocean, partially offset by an overall weakening in the large scale monsoon circulation (Turner and Annamalai, 2012; Sharmila et al., 2015; Sooraj et al., 2015). However, there is large uncertainty about the relative magnitude of these opposing effects, with several researchers predicting an overall weakening of the monsoon circulation based on alternative modelling experiments (e.g. Ashfaq et al., 2009; Krishnan et al., 2013, 2015).

In order to improve model predictions there is a need to study processes which are known to influence the behaviour of the monsoon but which are currently poorly understood (Knutti and Sedláček, 2012). One of the main sources of uncertainty concerns

interactions between the land surface and the atmosphere (Saha et al., 2011; Turner and Annamalai, 2012; Unnikrishnan et al., 2015). The lack of a common land use/land cover change dataset with which to force models has been identified as a major contributor to uncertainty regarding the influence of the land surface on precipitation (Pitman et al., 2009). Individual models follow different assumptions about the physical characteristics of specific land use/land cover types, resulting in strong discrepancies between model predictions (Pitman et al., 2009). While a number of global and regional land use/land cover products based on remote sensing are available (e.g. Hansen et al., 2000; Loveland et al., 2000; Friedl et al., 2002), none provide information about historical land use/land cover change in India as a result of the green revolution. Global datasets such as the History Database of the Global Environment (HYDE) fail to account for spatial heterogeneity in the rate of change because they are based on the assumption that the relative spatial distribution of land use/land cover remains constant. A further source of uncertainty arises from the different land surface schemes implemented by different climate models (Pitman et al., 2009), which is exacerbated by the fact that these models cannot be calibrated to local conditions. This is particularly relevant in northern India where processes such as irrigation, which until very recently have not been represented in Earth system models, are known to be important (Boucher et al., 2004). It has been shown that accurate modelling of soil moisture over India is essential to reliably simulate and forecast the South Asian monsoon (Saha et al., 2011; Asharaf et al., 2012; Unnikrishnan et al., 2015). To overcome this problem global climate models can be forced with the output of offline, high resolution, physics-based hydrological models, calibrated against local observed data (Seneviratne et al., 2010). The lack of such data for northern India is a major impedimant to scientific progress.

The research presented in this thesis forms part of Hydroflux India; a large, interdisciplinary project jointly funded by the UK Natural Environment Research Council and India's Ministry of Earth Sciences (MoES) to address the scientific challenges outlined above. Specifically, the project aims to quantify hydrometeorological feedbacks and changes in water storage and fluxes in northern India as a result of changes in land use, land cover and land management using a suite of models illustrated in Figure 1.2. This thesis initially focuses on developing an integrated modelling systems to generate input datasets for the various models used in Hydroflux India. The geographical focus of the work is on the Indian subcontinent but the methods developed, while optimised for Indian conditions, are designed to be applicable to many regional applications. It then considers the limitations of the current paradigm of hydrological and Earth system modelling for addressing the range of problems encountered during the Hydroflux India project, especially managing complex scientific workflows involving coupled human-environment systems.

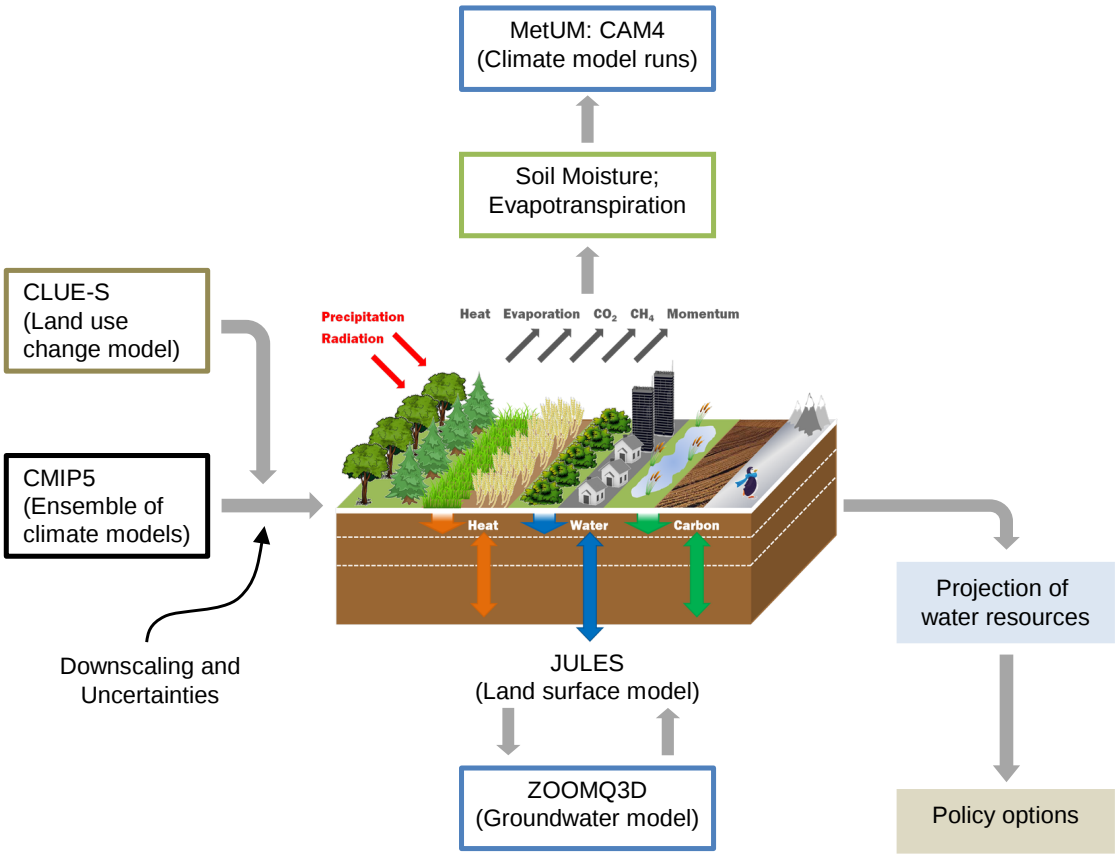


Figure 1.2.: Hydroflux India model setup. The initial focus of this thesis is on the development of appropriate tools and methods to generate the input datasets for the various models shown in the vertical direction. Then, it addresses the limitations of the current paradigm of Earth system modelling for complex scientific workflows involving coupled human-environment systems.

1.3 Aim and objectives

1.3.1 Aim

The aim of the thesis is to develop integrated modelling systems to improve the quantification of large-scale environmental change on regional water resources and climate.

1.3.2 Objectives

To achieve the stated aim the following objectives will be pursued:

1. Review the current state of the art in land use change modelling and develop an appropriate framework to develop historical land use/land cover maps at regional scales;

2. Apply the modelling framework to produce an historical land use/land cover change reconstruction for India;

3. Explore and develop methods to incorporate information about land management, particularly irrigation, with land use/land cover data, and develop a dataset showing the growth in irrigated area in India as a result of the green revolution;

4. Use the land use/land cover change dataset as an input to a regional land surface model simulation for northern India, and develop a post-processing algorithm that modifies simulated soil moisture values to include the effects of irrigation;

5. Review the limitations of current hydrological modelling practices for incorporating human-environment interactions, and explore the use of a data processing langauge for coupling models and orchestrating complex scientific workflows.

1.4 Thesis outline

This thesis is divided into seven chapters. The present Chapter describes the context of the thesis and its contribution to the Hydroflux India project and defines the aim and objectives that have guided the research.

Chapter 2 critically reviews the current state of the art in spatially explicit land use change modelling software. A new software framework which addresses the various identified limitations of the current modelling paradigm is proposed. The software implementation follows an object-oriented design intended to improve the efficiency of modelling workflows and facilitate ensemble model experiments and model intercomparison while promoting open and reproducible science. Its key features are demonstrated through an example application to simulate land use change in the Plum Islands Ecosystems site, a data-rich research site in the United States. The strengths and weaknesses of the framework are discussed and areas for development and improvement are identified.

Chapter 3 explains the importance of spatially explicit historical land use/land cover data for assessments of global and regional change and evaluates the available datasets for India. Considering the identified limitations, a new methodology is proposed whereby the Change in Land Use and its Effects (CLUE) land use change model, implemented in the software framework developed in Chapter 2, is used to spatially disaggregate district-level inventory data for India obtained from various sources. The resulting dataset provides annual land use/land cover maps for the Indian subcontinent between 1956–2010.

Chapter 4 builds on the work described in Chapter 3 by adapting the land use/land cover change dataset to show the expansion of irrigated land in India since the middle of

the twentieth century. Here, district-level agricultural inventory data is assigned to maps of cropland extent based on suitability maps obtained from the Food and Agriculture Organization Global Agro-Ecological Zones database. The dataset, which has the same spatial and temporal extent and resolution as the dataset developed in Chapter 3, shows the change in irrigated area of various crops during the study period.

Chapter 5 forces a process-based land surface model, the Joint UK Land and Environment Simulator (JULES), with the land use/land cover dataset developed in Chapter 3 to reconstruct historical soil moisture values for the Gangetic plains in northern India. A post-processing algorithm is applied to the JULES output to account for the effects of irrigation on soil moisture. The irrigated area dataset developed in Chapter 4, modified to account for the growing season of individual crops, is used to define the extent of irrigated land at each time point. Thus, two historical soil moisture datasets are produced: one considering land use/land cover change only, and one which additionally considers the effects of irrigation. These are used by other members of the Hydroflux India consortium to force two regional climate models to isolate the effects of irrigation on the behaviour of the South Asian monsoon.

Chapter 6 explores in more detail the development of integrated modelling systems to include anthropogenic activities in hydrological models. It first reviews the state of the art in hydrological modelling and modelling of coupled human-environment systems, before examining current technical solutions to model coupling and workflow orchestration. To explore the various issues further a data processing langauge, R, is set up as a flexible tool for workflow orchestration. It is supported through the development of a new software package which provides a set of classes to represent hydrological data and form the basis of a common interface for different models. The experimental set-up is demonstrated through two example applications focussing on hydrological data processing and model coupling, respectively.

Finally, Chapter 7 draws together the research and highlights the main contributions of the thesis in the context of the stated aim and objectives. Future areas of research and model development, particularly with respect to coupled human-environment systems, are highlighted.

2 Land use change modelling

Spatially explicit, historical data about land use/land cover change over time are an essential input to assessments of environmental change. While satellite data is commonly used to map land use/land cover in the recent past, the quality and availability of satellite images over India before the year 2000 is poor. Land use change models, which allow contemporary land use/land cover data to be extrapolated backwards or forwards in time based on statistical analysis of the relationship between the present-day landscape configuration and various socio-economic and biophysical covariables, provide an alternative approach. This chapter reviews the state of the art in land use change modelling software and identifies the following key limitations:

1. The source code for model implementations is frequently unavailable, compromising the reproducibility of scientific results and prohibiting members of the community from improving or adapting models for their own purposes;

2. Typical modelling workflows necessitate the use of additional software because existing applications usually only perform the spatial allocation of change;

3. It is difficult to perform ensemble experiments to capture model structural uncertainty because of fundamental differences between implementations of alternative models.

To address these issues a new software package, *lulcc*, is developed. The key features of the software are demonstrated through an example application to simulate land use change at the Plum Island Ecosystem Site in Massachusetts, USA. This dataset is convenient for demonstration purposes because it contains three land use/land cover maps for 1985, 1991 and 1999 and, since it has been the subject of several land use change modelling exercises, is familiar to the wider community.

Material presented in this chapter is adapted from the work described in Moulds, S., Buytaert, W. and Mijic, A. (2015). An open and extensible framework for spatially explicit land use change modelling: the lulcc R package. *Geoscientific Model Development*, 8:3215-3229, doi:10.5194/gmd-8-3215-2015.

2.1 Literature review

Spatially explicit land use change models are used to understand and quantify key processes that affect land use/land cover change and simulate past and future change (Veldkamp and Lambin, 2001; Mas et al., 2014). Land use/land cover change is the result of complex interactions between different biophysical and socio-economic conditions that vary across space and time (Verburg et al., 2002; Overmars and Verburg, 2006; Overmars et al., 2007). Several different model structures have been devised to capture this complexity and meet different objectives. Some models operate at global or regional scales to estimate the quantity of change at national or subnational levels based on economic considerations (e.g. Souty et al., 2012), whereas spatially explicit models, the focus of this chapter, operate over a spatial grid to predict the allocation of change (Mas et al., 2014). Inductive spatially explicit models are based on statistical models that predict the suitability of each model grid cell to a particular category (e.g. forest, grassland) or transition (e.g. forest $\rightarrow$ crop) as a function of spatially explicit covariables, while deductive models predict the allocation of change according to specific theories about the underlying processes (Overmars et al., 2007; Magliocca and Ellis, 2013). Inductive and deductive models operating at different spatial scales may be combined to better represent the complexity of a system (e.g. Castella and Verburg, 2007; Moreira et al., 2009). The main output of land use change models is a set of land use/land cover maps depicting the location of change over time. Detailed reviews of different models and modelling approaches are available in Verburg and Veldkamp (2004), Brown et al. (2013) and Mas et al. (2014).

Land use change models are commonly implemented in compiled languages such as C/C++ and Fortran and distributed as software packages or extensions to proprietary geographic information systems such as ArcGIS or IDRISI. As Rosa et al. (2014) point out, it is uncommon for the source code of land use change modelling software to be made available (e.g. Verburg et al., 2002; Soares-Filho et al., 2002; Verburg and Overmars, 2009; Schaldach et al., 2011). While it is true that the concepts and algorithms implemented by the software are normally described in scientific journal articles, this fails to ensure the reproducibility of scientific results (Peng, 2011; Morin et al., 2012), even in the hypothetical case of a perfectly described model (Ince et al., 2012). In addition, running binary versions of software makes it difficult to detect silent faults (faults that change the model output without obvious signals), whereas these are more likely to be identified if the source code is open (Cai et al., 2012). Moreover, it forces duplication of work and makes it difficult for members of the scientific community to improve the model code or adapt it for their own purposes (Morin et al., 2012; Pebesma, 2012; Steiniger and Hunter, 2013).

Current software packages for land use change modelling usually exist as specialised

applications that implement one spatial allocation algorithm. Indeed, it is common for applications to perform only one part of the modelling process. For example, the Change in Land Use and it Effects at Small regional extent (CLUE-S) software only performs the spatial allocation of change, requiring the user to prepare model input files and conduct the statistical analysis upon which the allocation procedure depends elsewhere (Verburg et al., 2002). This is time consuming and increases the likelihood of user errors because inputs to the various modelling stages must be transferred manually between applications. Furthermore, very few programs include methods to validate model output, which could be one reason for the lack of proper validation of models in the literature, as noted by Rosa et al. (2014). The lack of a common interface amongst land use change models is problematic for the community because there is widespread uncertainty about the appropriate model form and structure for modelling applications (Verburg et al., 2013). Under these circumstances it is useful to experiment with various models to identify the model that performs best in terms of calibration and validation (Schmitz et al., 2009). Alternatively, ensemble modelling may be used to understand the impact of structural uncertainty on model outcomes (Knutti and Sedláček, 2012). However, while some land use change model comparison studies have been carried out (e.g Pérez-Vega et al., 2012; Mas et al., 2014; Rosa et al., 2014), fundamental differences between models in terms of scale, resolution and model inputs prevent the widespread use of ensemble land use/land cover change predictions (Rosa et al., 2014). As a result, the uncertainty associated with the outcomes of land use change modelling studies is rarely communicated in a formal way, raising questions about the utility of such models to scientists and policymakers who need to have an appropriate level of confidence in model outcomes in order to make effective decisions (Pontius and Spencer, 2005; van Vliet et al., 2016).

An alternative approach is to develop frameworks that allow several modelling approaches to be implemented within the same environment. One such application is PCRaster, a free and open source GIS that includes additional capabilities for spatially explicit dynamic modelling (Schmitz et al., 2009). The PCRcalc scripting language and development environment allows users to build models with native PCRaster operations such as map algebra and neighbourhood functions. Alternatively, the PCRaster application programming interface (API) allows users to extend its functionality in various programming languages using native and external data types (Schmitz et al., 2009). For example, the current version of FALLOW (van Noordwijk, 2002; Mulia et al., 2014), a deductive land use change model, is built using the PCRaster framework. TerraME (Carneiro et al., 2013) is a platform to develop models for simulating interactions between society and the environment. It provides more flexibility than PCRaster because models can be composed of coupled sub-models with various temporal and spatial resolutions (Moreira et al., 2009; Carneiro et al., 2013). The platform is built on the open source TerraLib geospatial library

(Câmara et al., 2008), which handles several spatio-temporal data types, includes an API for coupling the library with R (R Core Team, 2016) to perform spatial statistical analysis, and supports dynamic modelling with cellular automata. The LuccME extension to TerraME includes implementations of CLUE-S and its predecessor, CLUE (Veldkamp and Fresco, 1996b; Verburg et al., 1999), written in the Lua programming language. However, while frameworks such as PCRaster and Terra-ME provide additional flexibility compared with specialised software packages, they lack functionality for data processing, statistical analysis and visualisation. Thus, users must still use additional software environments to achieve various ancillary tasks.

The R environment is a free and open source implementation of the S programming language; a language designed for programming with data (Chambers, 2008). Although the development of R is strongly rooted in statistical software and data analysis, it is increasingly used for dynamic simulation modelling in diverse fields (Petzoldt and Rinke, 2007). Additionally, in the last decade it has become widely used by the spatial analysis community, largely due to the *sp* package (Pebesma and Bivand, 2005; Bivand et al., 2013) which unified many alternative approaches for dealing with spatial data in R and allowed subsequent package developers to use a common framework for spatial analysis. The *raster* package (Hijmans, 2014) provides many functions for raster data manipulation commonly associated with GIS software. Building on these capabilities, several R packages have been created for dynamic, spatially explicit ecological modelling (e.g. Petzoldt and Rinke, 2007; Fiske and Chandler, 2011). In addition, two recent land use change models have been written for the R environment. StocModLCC (Rosa et al., 2013) is a stochastic inductive land use change model for tropical deforestation while SIMLANDER (Hewitt et al., 2013) is a stochastic cellular automata model to simulate urbanisation. Thus, R is well suited to spatially explicit land use change modelling. To date, however, it has not been used to develop a framework for land use change model development and comparison. The remainder of this chapter is divided into three sections. Firstly, the software is described and its main functionality is demonstrated using an example application to simulate land use change at the Plum Island Ecosystems site in Massachusetts, United States. This is followed by a discussion of the strengths and main limitations of the software and the general approach, as well as areas for future development. Lastly, brief conclusions from the project are drawn.

2.2 Software description

The *lulcc* package follows an object-oriented design. Object-oriented programming is based on the concepts of data abstraction, encapsulation, inheritance and polymorphism

(Lewallen, 2005). Encapsulation is the act of concealing the implementation of data in objects so that the user of the data does not need to be concerned with where, and in what form, the data is stored. Data abstraction is the notion of reducing a real-world object to its essential characteristics for a given application in order to reduce complexity and improve efficiency. Inheritance allows new class definitions to include the properties of existing classes, while polymorphism is the idea that a given method has different implementations for objects belonging to different classes. The design of *lulcc* provides a formal structure for the modelling framework which allows the various stages of land use change modelling applications, illustrated in Figure 2.1, to be handled efficiently. Furthermore, it encourages the reuse of code because objects can be used multiple times within the same application or across several different applications. It is extensible because it is straightforward to extend existing classes or create new methods for existing classes. The package uses the S4 class system for object-oriented programming in R (Chambers, 1998, 2008), which requires classes and methods to be formally defined. This system is more rigorous than the alternative S3 system because objects are validated against the class definition when they are created, ensuring that objects behave consistently when they are passed to functions and methods. Figure 2.2 shows the relationship between the RasterStack class of package *raster* and the three classes representing spatial data in *lulcc*, while Figure 2.3 shows the overall structure of the software package. Table 2.1 summarises the functions included with the package.

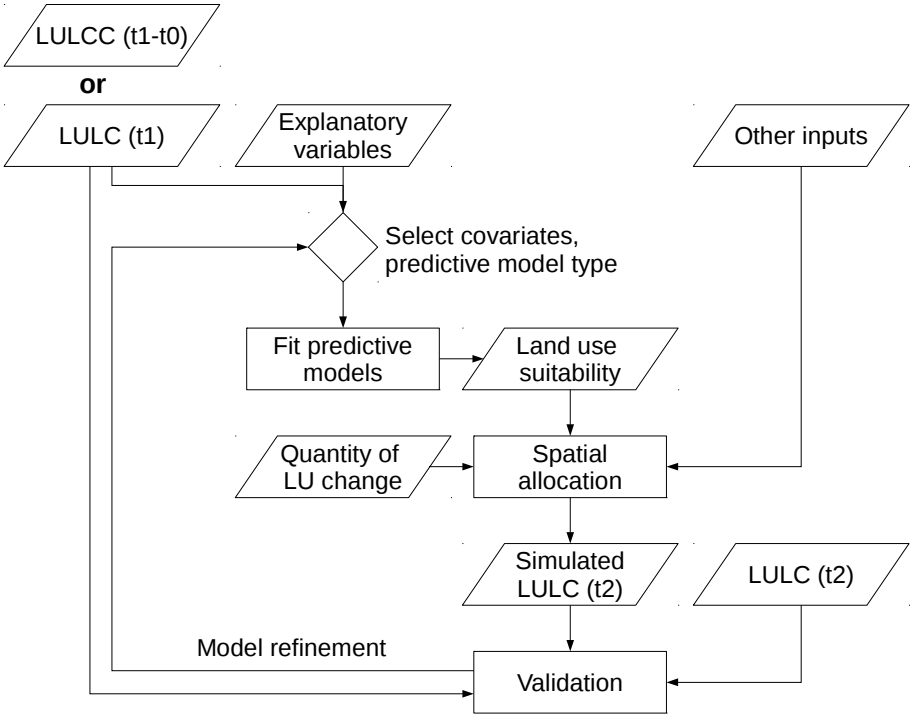


Figure 2.1.: Diagram showing the general methodology used for inductive land use change modelling applications, adapted from Mas et al. (2014). The input land use/land cover data can be a single categorical map showing the pattern of land use/land cover at one time point (LULC (t1)) or a series of maps showing historical land use/land cover transitions (LULCC (t1-t0)).

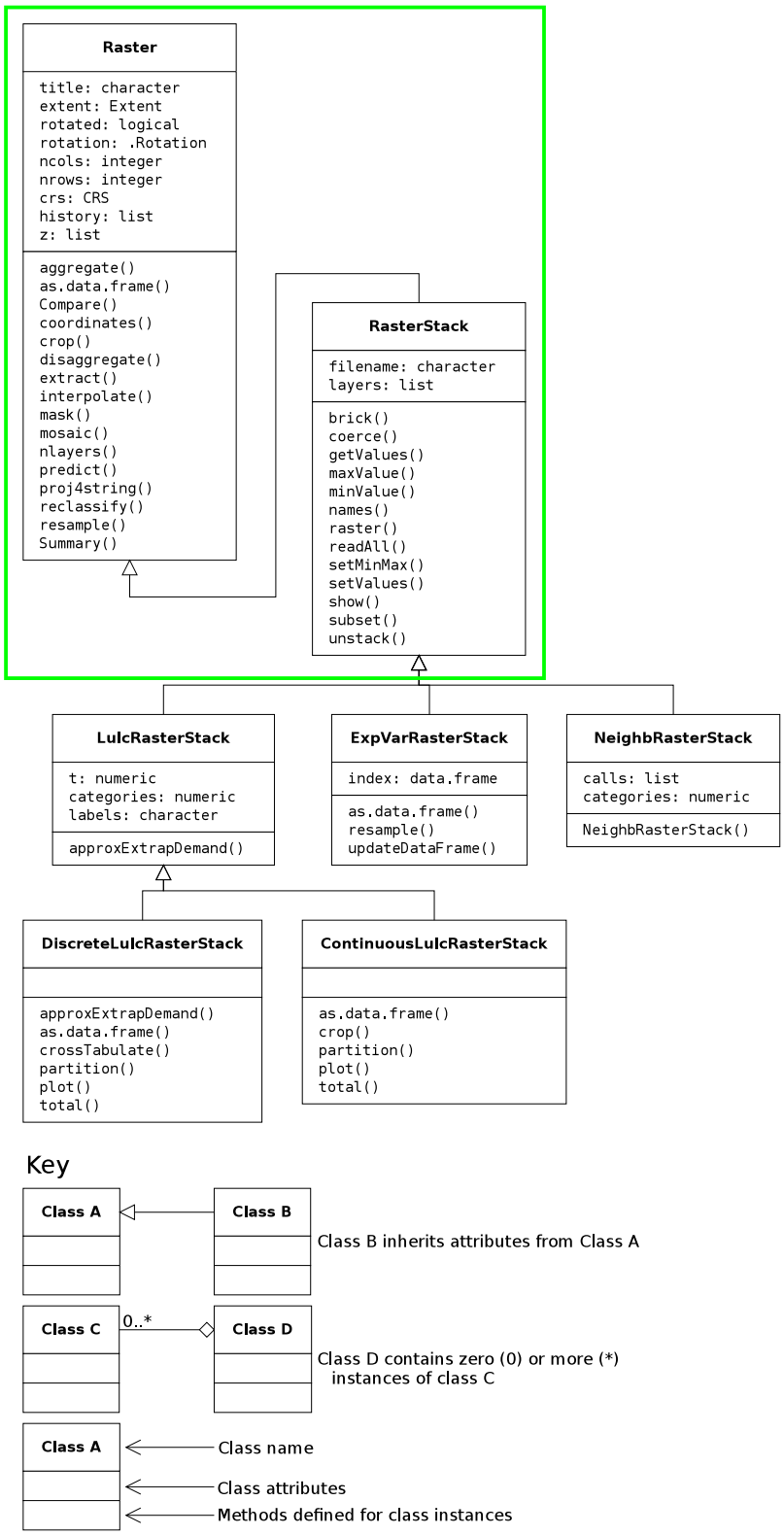


Figure 2.2.: Class diagram in the Unified Modeling Language (UML) showing the relationship between *raster* and *lulcc* classes. Classes defined in package *raster* are shown inside the green box. Only a selection of methods are shown due to the very large number of methods for manipulating raster data defined in package *raster*.

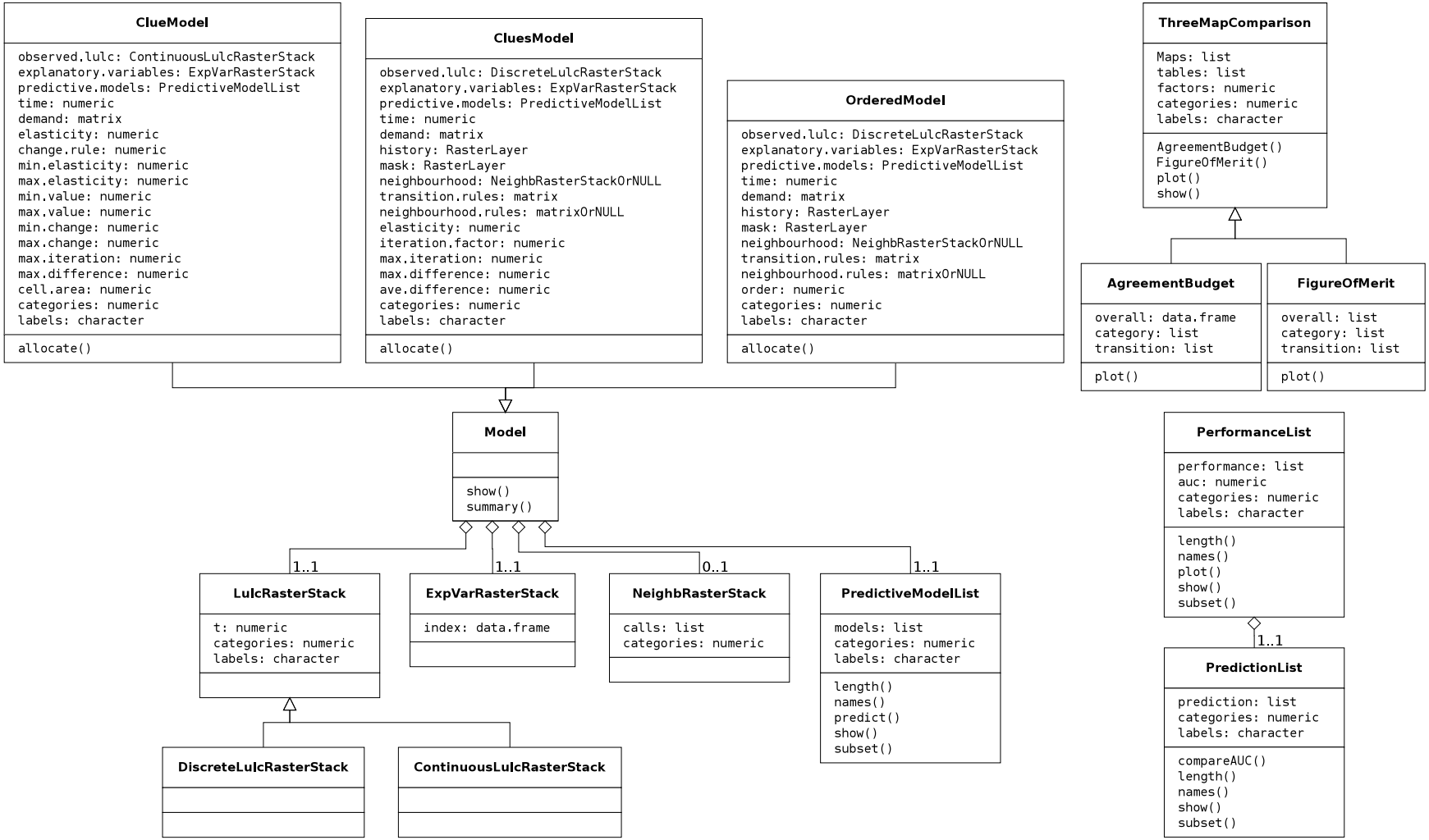


Figure 2.3.: Class diagram in the Unified Modeling Language (UML) for *lulcc*, showing the main classes and methods included in the package.

In the remainder of this section the main features of *lulcc* are demonstrated through an example application to the Plum Island Ecosystems site in Massachusetts, USA, using a dataset included with the package. The study area, which has been widely investigated by the land use change modelling community (e.g. Schneider et al., 2001; Pontius and Li, 2010; Pontius and Parmentier, 2014; Bradley et al., 2016), provides a familar reference point for potential users of the software. Moreover, the fact that a large amount of data is available for site makes it easier to demonstrate the key functionality of the software. The land use/land cover maps in the Plum Island Ecosystems dataset are discrete maps in which each grid cell is entirely devoted to exactly one land use/land cover category.

2.2.1 Data

The failure to provide driving data for land use change modelling studies alongside published literature is a major weakness of the discipline (Rosa et al., 2014). The *lulcc* package includes two datasets which allow users to quickly start exploring the modelling framework using real-world case studies that have been widely studied by the community. The first of these contains data from the Plum Island Ecosystems Long Term Ecological Research site in northeast Massachusetts, which in recent decades has undergone extensive land use change from forest to residential use (Aldwaik and Pontius, 2012). The dataset included in *lulcc* was originally developed as part of the MassGIS program (MassGIS, 2015) but has subsequently been processed by Pontius and Parmentier (2014). Land use/land cover maps depicting forest, residential and other uses are available for 1985, 1991 and 1999 together with maps of three predictor variables: elevation, slope and distance to built land in 1985. The second dataset is a modified version of the Sibuyan Island dataset supplied with the original CLUE-S model implementation (Verburg et al., 2002). This allows users to compare the *lulcc* version of CLUE-S with the original implementation using a common dataset. The remainder of this section focuses on the Plum Island Ecosystem dataset.

2.2.2 Data processing

One of the most challenging aspects of land use change modelling is to obtain and process input data (Pontius and Malanson, 2005). Currently, *lulcc* requires all spatially explicit input data to exist either in the file system or in the R workspace as *raster* objects (RasterLayer, RasterStack or RasterBrick). The most fundamental input required by land use change models is an initial map of observed land use/land cover, which is usually obtained from classified remotely sensed data. This map represents the initial condition for model simulations and, for inductive modelling, is used to fit predictive models. Sometimes it is more useful to consider observed transitions (Brown et al., 2002): in this case an additional

Table 2.1.: Functions included in the *lulcc* package. Constructor functions for the various classes shown in Figure 2.3 are not shown.

Function name	Description
allocate	Perform spatial allocation using various methods
approxExtrapDemand	Create a demand scenario by linear extrapolation
as.data.frame	Coerce *lulcc* objects to data frames
c.PredictiveModelList	Concatenate PredictiveModelList objects
clue	Run CLUE allocation algorithm for one time step
clues	Run CLUE-S allocation algorithm for one time step
compareAUC	Compare the area under the curve (AUC) of various predictive models
crossTabulate	Calculate the contingency table for two categorical raster maps
getPredictiveModelInputData	Create a data frame containing variables required to fit predictive models
length	Get the length of a *lulcc* object
ordered	Run ordered allocation algorithm for one time step
partition	Divide a Raster map into training and testing partitions
predict	Make predictions using a PredictiveModelList object
resample	Resample an ExpVarRasterStack object or a list of Raster objects to the parameters of another Raster object
roundSum	Round elements in a matrix or data frame ensuring that all rows sum to the same value
show	Display the characteristics of a *lulcc* object
subset	Subset elements in *lulcc* objects
summary	Summarise *lulcc* objects
total	Sum the total number of cells belonging to each class of a categorical raster map
updateDataFrame	Update data frame containing dynamic variables for new time point

map for an earlier time point is required, as shown in Figure 2.1. Ideally, two additional maps for subsequent time points should be obtained for calibrating and validating the land use change model (Pontius et al., 2004a).

Land use/land cover data in *lulcc* are represented by the LulcRasterStack class. Derived classes DiscreteLulcRasterStack and ContinuousLulcRasterStack inherit attributes from this class and support land use/land cover maps with discrete data, in which each pixel represents exactly one category, and continuous data, in which each pixel has fractional membership to all the categories in the study region, respectively. In the following code snippet the *lulcc* package is loaded into the current R session and a DiscreteLulcRasterStack object for the Plum Island Ecosystems dataset is created:

```
devtools::install_github("simonmoulds/r_lulcc2/lulcc")
data(pie)
lu <- DiscreteLulcRasterStack(x=stack(pie[1:3]),
                              categories=c(1,2,3),
                              labels=c("Forest","Built","Other"),
                              t=c(0,6,14))
plot(lu)
```

The resulting plot, with some refinements, is shown in Figure 2.4. The LulcRasterStack object is important to land use change modelling workflows using *lulcc* because it defines the spatial domain of subsequent operations. The `t` argument in the constructor function specifies the time points associated with the observed maps. In most land use change modelling applications the timestep between two time points represents one year but there is no requirement for this to be the case.

A common starting point in land use change modelling is to obtain a transition matrix for observed maps from two time points to identify the main historical transitions in the study region (Pontius et al., 2004b). This can be used as the basis for further research into the processes driving change. In *lulcc* the `crossTabulate` function is used for this purpose:

```
crossTabulate(x=lu, times=c(0,14))

##        Forest Built Other
## Forest  44107  4250   656
## Built      11 36957   154
## Other    1259  2248 23921
```

The transition matrix reveals that for the Plum Island Ecosystems site the dominant change

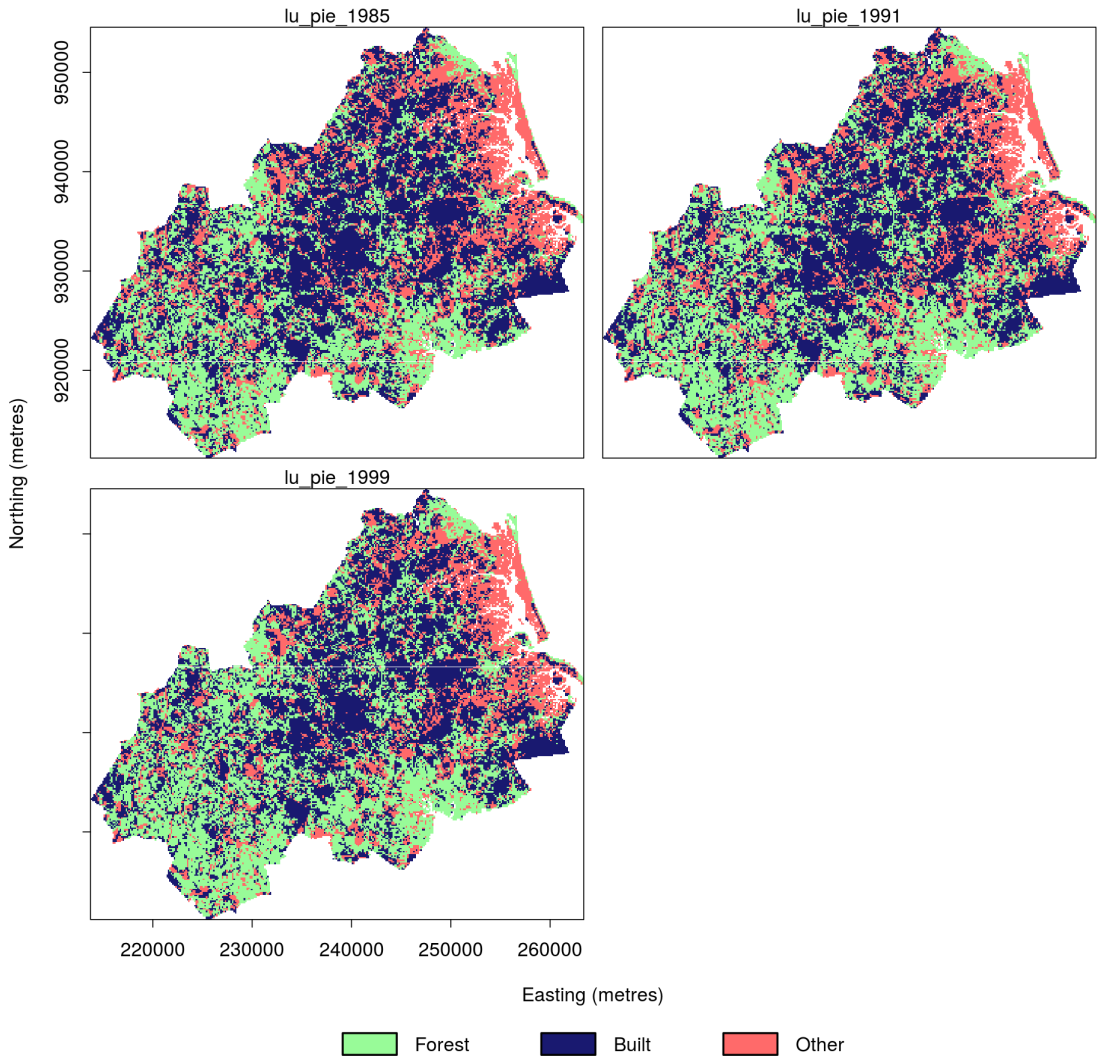


Figure 2.4.: Observed land use/land cover maps for the Plum Island Ecosystems site in 1985, 1991 and 1999 created by plotting the LulcRasterStack object representing the data. The maps are projected in the Lambert Conformal Conic projection system.

between 1985 and 1999 was the conversion of forest to built areas.

Inductive and deductive land use change models predict the allocation of change based on spatially explicit biophysical and socio-economic explanatory variables. These may be static, such as elevation or geology, or dynamic, such as maps of population density or road networks. In *lulcc*, explanatory variables are collectively represented by an object of class ExpVarRasterStack. To identify whether an explanatory variable is static or dynamic a data frame is supplied to the ExpVarRasterStack constructor function with three columns: the first column specifies the name of the variable, the second column specifies the time point for which it is relevant and the third column specifies whether it is dynamic or not. Data frame rows should correspond to the individual layers of the RasterStack object containing the explanatory variables. The ExpVarRasterStack object for Plum Island Ecosystems is created as follows:

```
idx <- data.frame(var=c("ef_001","ef_002","ef_003"),
                  yr=c(0,0,0),
                  dynamic=c(FALSE,FALSE,FALSE))
idx

##      var yr dynamic
## 1 ef_001  0   FALSE
## 2 ef_002  0   FALSE
## 3 ef_003  0   FALSE

ef <- ExpVarRasterStack(x=stack(pie[4:6]), index=idx)
```

Where variables ef_001, ef_002 and ef_003 correspond to elevation, slope and distance to built land in 1975, respectively. Apart from maps of observed land use/land cover and explanatory variables, two other input maps may be required. The allocation routines currently implemented by *lulcc* accept a mask file, which is used to prevent change within a certain geographic area such as a national park or other protected area, and a land use/land cover history file, which is used as the basis for certain decision rules. These maps are handled by *lulcc* as standard RasterLayer objects. All input maps should have the same spatial resolution as the corresponding LulcRasterStack object. This can be achieved using the `resample` function from the *raster* package. For example, the ExpVarRasterStack object created above can be resampled to the parameters of the corresponding LulcRasterStack object as follows:

```
ef <- resample(ef, lu)
compareRaster(ef, lu)

## [1] TRUE
```

2.2.3 Predictive modelling

Inductive land use change models relate the pattern of observed land use/land cover to spatially explicit explanatory variables. Logistic regression is a common type of predictive model used for inductive land use change modelling (e.g. Pontius and Schneider, 2001; Verburg et al., 2002). However, there is growing interest in the application of local and non-parametric models (e.g. Tayyebi et al., 2014). One reason why R is attractive for land use change modelling is that it has become the *de facto* standard for statistical software development. As a result, *lulcc* can easily support various predictive modelling techniques through existing R packages. For discrete land use/land cover data, *lulcc* currently supports binary logistic regression, available in base R, recursive partitioning and regression trees, provided by the *rpart* package (Therneau et al., 2014), and Random Forests (Breiman, 2001), provided by the *randomForest* package (Liaw and Wiener, 2002). In this case, the statistical models predict the suitability of each cell to the respective land uses. For continuous data the package supports linear regression as well as recursive partitioning and regression trees and Random Forests. Here, the models predict the fraction of each cell belonging to the various land uses.

Parametric models such as logistic regression assume the data to be independent and identically distributed (Overmars et al., 2003). However, in spatial analysis the presence of spatial autocorrelation, which reduces the information content of an observation because its value can to some extent be predicted by the value of its neighbours, often results in this assumption being violated (Beale et al., 2010). There is some evidence that non-parametric models such as Random Forests may be affected by spatial autocorrelation even though they do not assume independence Mascaro et al. (2014). An approach to reduce the impact of this phenomenon is to fit predictive models to a random subset of the data (e.g. Verburg et al., 2002; Wassenaar et al., 2007; Echeverria et al., 2008). To fit predictive models to discrete land use/land cover data, the dataset is first divided into training and testing partitions by performing a stratified random sample of the grid cells in the study area. Data points in the training partition are then extracted from the various maps using the `getPredictiveModelInputData` function and the resulting data frame is used to fit predictive models for each category:

```
part <- partition(x=lu, size=0.1, spatial=TRUE, t=0)

## Note:  the specification for S3 class "family" in package 'MatrixModels' seems
equivalent to one from package 'lme4':  not turning on duplicate class definitions
for this class.

train.data <- getPredictiveModelInputData(lu=lu,
                                          ef=ef,
                                          cells=part[["train"]],
                                          t=0)


forest.form  <- as.formula("Forest ~ ef_001 + ef_002")
built.form   <- as.formula("Built ~ ef_001 + ef_002 + ef_003")
other.form   <- as.formula("Other ~ ef_001 + ef_002")

forest.glm   <- glm(forest.form, family=binomial, data=train.data)
forest.rprt  <- rpart(forest.form, data=train.data)
forest.rf    <- randomForest(forest.form, method="class", data=train.data)

built.glm    <- glm(built.form, family=binomial, data=train.data)
built.rprt   <- rpart(built.form, data=train.data)
built.rf     <- randomForest(built.form, method="class", data=train.data)

other.glm    <- glm(other.form, family=binomial, data=train.data)
other.rprt   <- rpart(other.form, data=train.data)
other.rf     <- randomForest(other.form, method="class", data=train.data)
```

Subsequently the various predictive models are grouped into an object of class PredictiveModelList:

```
glm.mods   <- PredictiveModelList(list(forest.glm, built.glm, other.glm),
                                  categories=c(1,2,3),
                                  labels=c("Forest","Built","Other"))

rprt.mods <- PredictiveModelList(list(forest.rprt, built.rprt, other.rprt),
                                 categories=c(1,2,3),
                                 labels=c("Forest","Built","Other"))

rf.mods    <- PredictiveModelList(list(forest.rf, built.rf, other.rf),
                                  categories=c(1,2,3),
                                  labels=c("Forest","Built","Other"))
```

Methods to evaluate statistical models are provided by the *ROCR* package (Sing et al., 2005). This allows users to assess model performance using various methods such as the receiver operator characteristic (ROC), which is commonly used to measure the performance of models predicting the presence or abscence of a phenomenon (Pontius and Parmentier, 2014). The ROC curve is often summarised by the area under the curve (AUC), where one indicates a perfect fit and 0.5 indicates a purely random fit. In *lulcc* the native *ROCR* classes are extended to better suit the task of simultaneously assessing multiple predictive models. Classes "prediction" and "performance" are extended by PredictionList and PerformanceList, respectively, to handle objects of class PredictiveModelList. In the following example the logistic regression models fitted above are evaluated using the testing partition from the 1985 observed map:

```
test.data <- getPredictiveModelInputData(lu=lu,
                                         ef=ef,
                                         cells=part[["test"]],
                                         t=0)
glm.pred  <- PredictionList(models=glm.mods, newdata=test.data)
glm.perf  <- PerformanceList(pred=glm.pred, measure="rch")

rprt.pred <- PredictionList(models=rprt.mods, newdata=test.data)
rprt.perf <- PerformanceList(pred=rprt.pred, measure="rch")

rf.pred   <- PredictionList(models=rf.mods, newdata=test.data)
rf.perf   <- PerformanceList(pred=rf.pred, measure="rch")
```

The ROC curves for each land use/land cover category and for each type of predictive model are shown in Figure 2.5. The plots show that binary logistic regression and random

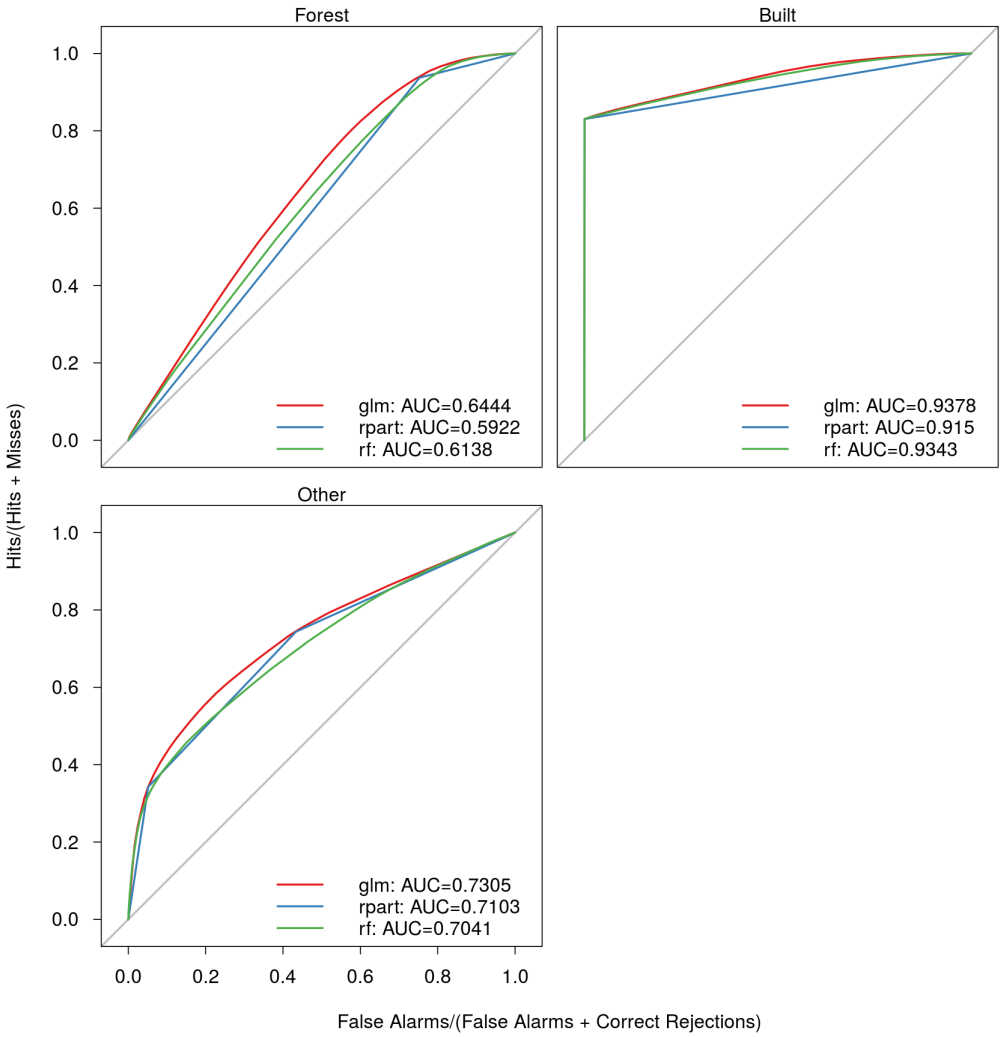


Figure 2.5.: ROC curves showing the ability of each type of predictive model to simulate the observed pattern of land use/land cover in the Plum Island Ecosystems site in 1985 in the data partition left out of the fitting procedure.

forest models perform similarly for all categories, while regression tree models perform least well.

Another use of ROC analysis is to assess how well the models predict the cells in which gain occurs between two time points, which is only possible if a second observed land use/land cover map is available for a subsequent time point. This analysis is performed below to assess the ability of the model for built land to predict the location of the gain of Built between 1985 and 1991. A data partition eliminating cells not candidate for gain (cells belonging to Built in 1985) is created and supplied to `getPredictiveModelInputData` to extract data points for the cells eligible to change to Built. The resulting data frame is used to assess the ability of the various predictive models to predict the gain of Built between 1985–1991, as follows:

```
nonbuilt <- rasterToPoints(lu[[1]],
                           fun=function(x) x != 2,
                           spatial=TRUE)
test.data <- getPredictiveModelInputData(lu=lu,
                                         ef=ef,
                                         cells=nonbuilt,
                                         t=6)
glm.pred <- PredictionList(models=glm.mods[2], newdata=test.data)
glm.perf <- PerformanceList(pred=glm.pred, measure="rch")
```

The ROC curve is shown in Figure 2.6.

The PredictiveModelList class makes it straightforward to map the suitability of every pixel in the study region to the various categories using the generic `predict` function with some additional functionality from package *raster*, as follows:

```
all.data <- as.data.frame(x=ef, cells=part[["all"]])
probmaps <- predict(object=glm.mods,
                    newdata=all.data,
                    data.frame=TRUE)
points   <- rasterToPoints(lu[[1]], spatial=TRUE)
probmaps <- SpatialPointsDataFrame(points, probmaps)
probmaps <- rasterize(x=probmaps, y=lu[[1]],
                      field=names(probmaps))
```

The resulting RasterStack object is shown in Figure 2.7.

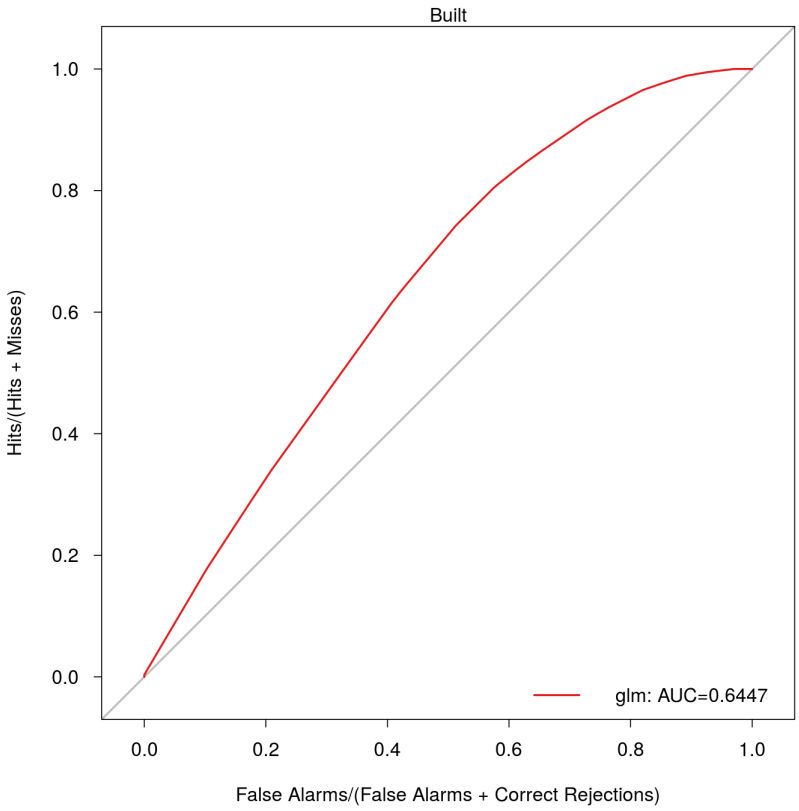


Figure 2.6.: ROC curve showing the ability of the binary logistic regression model fitted on observed land use/land cover data from 1985 to predict the gain in Built land between 1985 and 1991.

In some circumstances it may be appropriate to supply a model with no explanatory variables to an allocation routine. For example, Verburg and Overmars (2009) used such a model for natural and semi-natural vegetation because in their particular case study the selection of pixels for conversion to these categories was based on the suitability of pixels to agricultural and urban land rather than the suitability of natural and semi-natural vegetation. In *lulcc*, this can most easily be achieved by fitting a binary logistic regression model with no explanatory variables. To do this, a formula such as `Forest~1` should be supplied to the `glmModels` function.

2.2.4 Demand

Spatially explicit land use change models are usually driven by non-spatial estimates of either the total number of cells occupied by each category at each time point or the number of transitions among the various categories during each time interval. This means

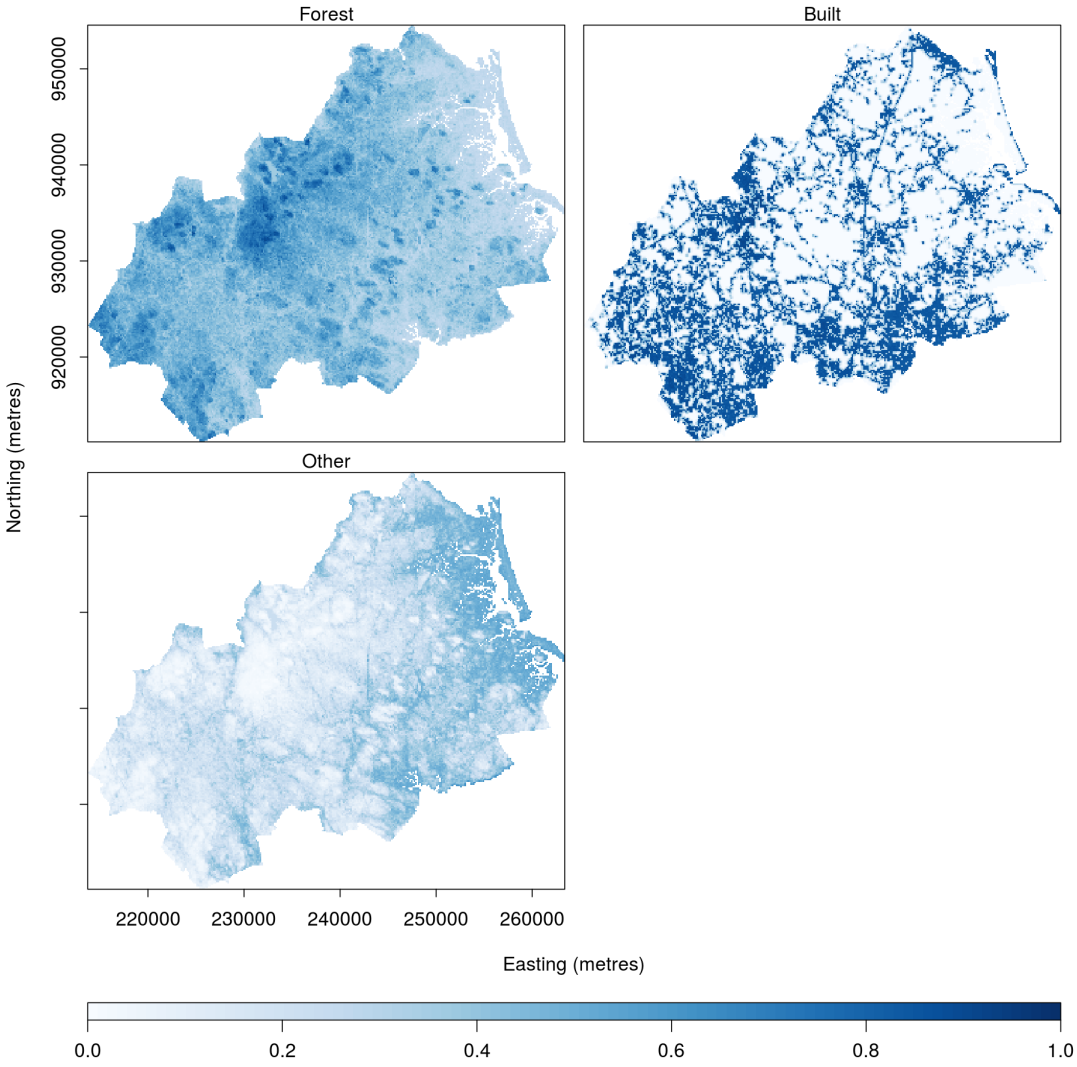


Figure 2.7.: Suitability of pixels in the Plum Island Ecosystems study site to belong to Forest, Built and Other land use classes according to binary logistic regression models. Elevation and slope are used as explanatory variables for all land uses while Built additionally includes distance to built pixels in 1985. The maps are projected in the Lambert Conformal Conic projection system.

regional drivers of land use/land cover change, such as population growth and technology, are considered implicitly because they influence the quantity of the respective land uses rather than necessarily the spatial allocation of land use change (Fuchs et al., 2013). While some models calculate demand at each time point based on the spatial configuration of the landscape at the previous time point (e.g. Rosa et al., 2013), it is more common to specify the demand for every time point at the beginning of the simulation (e.g. Pontius and Schneider, 2001; Verburg et al., 2002; Sohl et al., 2007). The way in which demand is specified in *lulcc* is unique to individual allocation models. Currently, however, both allocation models currently included in the package require the total number of cells belonging to each category at every time point to be supplied as a matrix or data frame before running the allocation routine.

The area of each land use/land cover category may be estimated using non-spatial land use models or, in the case of a backcast model, national and subnational land use/land cover inventory data (e.g. Ray and Pijanowski, 2010; Fuchs et al., 2013). The package includes a function to interpolate or extrapolate area based on two or more observed land use/land cover maps: this approach is often used to predict the quantity of change in the near-term (Mas et al., 2014). For the present application demand is obtained for each year between 1985 and 1999 by linear interpolation, as follows:

```
dmd <- approxExtrapDemand(lu=lu, tout=0:14)
```

Simulating land use/land cover change between two time points for which observed data is available, as above, is a useful exercise for model pattern validation because it allows models to be tested based on their ability to predict the spatial allocation of change given the exact quantity of change.

2.2.5 Allocation

The allocation algorithm in land use change models determines the grid cells in which various transitions should take place (Verburg et al., 2002). Currently, *lulcc* includes implementations of the CLUE model (Veldkamp and Fresco, 1996b; Verburg et al., 1999), which is designed for continuous land use/land cover data, as well as the CLUE-S model and a novel stochastic ordered procedure based on the algorithm described by Fuchs et al. (2013), both of which are designed for discrete land use/land cover data.

Decision rules

The CLUE-S and ordered allocation routines allow the user to optionally provide various decision rules which are implemented before the main allocation algorithm. The first decision rule included in *lulcc* is used to prohibit certain transitions. For example, in most situations it is unlikely that urban areas will be converted to agricultural land because the initial cost of urban development is high (Verburg et al., 2002). The second rule specifies a minimum number of timesteps before a certain transition is allowed, while the third rule specifies a maximum number of timesteps after which change is not allowed. These rules are used to control transitions that are time-dependent, such as the transition from shrubland to closed forest (Verburg and Overmars, 2009). The fourth rule prohibits transitions to a certain category in cells that are not within a user-defined neighbourhood of cells already belonging to that category. This rule is particularly relevant to cases of deforestation or urbanisation where change occurs at the forest or urban fringe.

Within the `allocate` function the first three decision rules are applied by the `allow` function and the fourth rule is applied by the `allowNeighb` function. For time dependent decision rules the user should supply a land use/land cover history raster map, specifying the length of time each pixel has belonged to the current category. If this is not supplied each pixel is assigned a value of one, representing one model timestep. To apply neighbourhood rules it is necessary to supply corresponding neighbourhood maps to the allocation routine. In *lulcc* these are represented by the `NeighbRasterStack` class. Objects of this class are created with the following command:

```
w  <- matrix(data=1, nrow=3, ncol=3)
nb <- NeighbRasterStack(x=lu[[1]], weights=w, categories=c(1,2,3))
```

The `allow` and `allowNeighb` functions identify disallowed transitions according to the respective decision rules and set the suitability of these cells to NA (not available). These transitions are ignored by the allocation routine. Care should be taken to ensure that after any decision rules are taken into account there are sufficient cells eligible to change in order to meet the specified demand at each time point.

CLUE-S allocation method

The CLUE-S model (Verburg et al., 2002) implements an iterative routine to meet the specified demand at each time point and handle competition between the various land use/land cover categories. The procedure is based on regression models which predict the probability, $P_{i,u}$, of land use/land cover u belonging to grid cell i based on various covariables. The original implementation of the CLUE-S algorithm requires the use of

binomial logistic regression. However, as discussed previously, *lulcc* supports binomial logistic regression, recursive partitioning and regression trees and Random Forests. The total suitability, $tprob_{i,u}$, is calulcated

$$tprob_{i,u} = P_{i,u} + elas_u + iter_u \tag{2.1}$$

where $elas_u$ is the relative elasticity to change and $iter_u$ is an iteration factor which initially has the same value for all categories. Relative elasticity to change varies between zero and one where zero represents high elasticity to change and one represents low elasticity to change. The relative elasticity term is only included in Equation 2.1 if grid cell i is under land use/land cover u in the year under consideration. At the beginning of the allocation procedure each grid cell is set to the category with the highest total suitability. The algorithm then determines whether the allocated area of each category is less than, equal to or greater than the specified demand. If it is less than or greater than demand the iteration factor for the category in question is increased or decreased, respectively, by an amount proportional to the difference between the allocated area and the demand. If the allocated area equals demand the suitability is left unchanged. Based on the new values of the iteration factor the total probability is recalculated and the allocation procedure is repeated. This iterative cycle continues until the demand for all categories, within a user-defined tolerance, is met or the maximum number of iterations is reached. The *lulcc* implementation of CLUE-S is based solely on the description of the algorithm provided by Verburg et al. (2002). As a result, users should not expect to reproduce exactly the output from the original model implementation, which is closed source, because the model description does not provide the necessary level of detail required for its unambiguous conversion to computer code.

In *lulcc* the data required to run specific allocation models are represented by classes inheriting from Model, as shown in the class diagram (Figure 2.3). The CLUE-S allocation model is represented by class CluesModel which, for the current study area, is created as follows:

```
clues.model <- CluesModel(observed.lulc=lu,
                          explanatory.variables=ef,
                          predictive.models=glm.mods,
                          time=0:14,
                          demand=dmd,
                          history=NULL,
                          mask=NULL,
                          neighbourhood=NULL,
                          transition.rules=matrix(data=1, nrow=3, ncol=3),
```

```
                                neighbourhood.rules=NULL,
                                elasticity=c(0.2,0.2,0.2),
                                iteration.factor=0.00001,
                                max.iteration=1000,
                                max.difference=5,
                                ave.difference=5)
```

To run the model the generic `allocate` function receives objects inheriting from Model and, depending on the subclass, implements the correct allocation algorithm:

```
clues.result <- allocate(clues.model)
```

The result is an object of class DiscreteLulcRasterStack containing a map for each year in the study period.

Ordered method

The ordered allocation method is based on the algorithm described by Fuchs et al. (2013). It is less computationally expensive and more stable than the CLUE-S algorithm because it does not simulate competition between land use/land cover categories. Instead, land allocation is performed in a hierarchical way according to the perceived socio-economic value of each category. For categories with increasing demand only cells belonging to categories with lower socio-economic value are considered for conversion. In this case, n cells with the highest suitability to the category under consideration are selected for change, where n equals the number of transitions required to meet the demand. The converted cells, as well as the cells that remain under the current category, are masked from subsequent operations. For categories with decreasing demand only cells belonging to that category are allowed to change. Here, n cells with the lowest allocation suitability are converted to a temporary class which can be allocated to subsequent categories. The category with the lowest socio-economic value is a special case because it is considered last and, therefore, the number of cells that have not been assigned to other categories must equal the demand.

In *lulcc* the algorithm described by Fuchs et al. (2013) is modified to allow stochastic transitions. If this option is selected, the allocation suitability of each cell allowed to change is compared to a random number between zero and one drawn from a uniform distribution. If demand for the land use/land cover category is increasing only cells where the allocation suitability is greater than the random number are allowed to change, whereas for decreasing demand only cells where it is less than the random number are allowed to change. To make the model deterministic the user can set the `stochastic` argument to FALSE when

the `allocate` function is called.

The ordered allocation model is represented by the OrderedModel class. In the following code an object of class OrderedModel is created, including the order in which to allocate change (built → forest → other), and then passed to the `allocate` function to execute the model:

```
ordered.model <- OrderedModel(observed.lulc=lu,
                              explanatory.variables=ef,
                              predictive.models=glm.mods,
                              time=0:14,
                              demand=dmd,
                              transition.rules=matrix(data=1, 3, 3),
                              order=c(2,1,3))
ordered.result <- allocate(ordered.model, stochastic=FALSE)
```

CLUE allocation method

The CLUE model (Veldkamp and Fresco, 1996a; Verburg et al., 1999) simulates land use/land cover change over a grid in which individual cells have partial membership to various categories. It was originally designed to simulate future land use/land cover change at national and regional levels, and case studies exist for Ecuador (Veldkamp and Fresco, 1996a), Costa Rica (Veldkamp and Fresco, 1996b) and Java, Indonesia (Verburg et al., 1999). In common with CLUE-S and the ordered allocation model, CLUE is based on statistical analysis of the quantitative relationship between the contemporary distribution of land use/land cover and selected socio-economic and biophysical covariates (Verburg et al., 1999). The resulting regression models, one for each land use/land cover category in the study region, are used to predict the fraction of the respective categories occupying each grid cell. This quantity, referred to as the regression cover, indicates the expected area of each category given the underlying biophysical and socio-economic conditions (Verburg et al., 1999). At each time point an iterative routine adjusts the regression cover in each cell until the total area of each category meets a specified demand. During this procedure, if the area of a particular category is increasing according to the demand scenario its fraction in grid cells where the regression cover is lower than the actual cover from the previous time point is increased by an amount proportional to the difference between the two values. These cells are prioritised for change because the underlying statistical model for the category under consideration indicates that cells under similar conditions located elsewhere have a higher fractions. From Verburg et al. (1999), the allocation of change in an individual grid cell can be expressed

$$cover_{i,t,u} = cover_{i,t-1,u} + [(regr_{i,t,u} - cover_{i,t-1,u}) \times iter_{i,u}] \quad (2.2)$$

where $cover_{i,t,u}$ is the fraction of land use/land cover u in grid cell i at time t, $cover_{i,t-1,u}$ is the fraction of land use/land cover u at the previous time point, $regr_{i,t,u}$ is the regression cover and $iter_u$ is an iteration factor which is adjusted until the allocated area of each category equals the specified area at the regional level, as follows

$$demand_{t,u} = \sum_{i=1}^{n}(cover_{i,t-1,u} + [(regr_{i,t,u} - cover_{i,t-1,u}) \times iter_{i,u}]) \quad (2.3)$$

where $demand_{t,u}$ is the total area of land use/land cover u at time t and n is the number of cells in the study region. The CLUE model is not demonstrated here because the Plum Island Ecosystems dataset contains discrete land use/land cover maps. Instead, it is used in Chapter 3 to reconstruct historical land use/land cover in India.

2.2.6 Pattern validation

Spatially explicit land use change models are validated by comparing the initial observed land use/land cover map with an observed and simulated map for a subsequent time point (Pontius et al., 2011). While previous studies have extracted useful information from the three possible two-map comparisons (e.g. Pontius et al., 2007), Pontius et al. (2011) devised the concept of a three-dimensional contingency table to compare the three maps simultaneously. Not only is this approach more parsimonious, it also yields more information about quantity and allocation performance (Pontius et al., 2011). For example, from the three-dimensional contingency table it is straightforward to identify sources of agreement and disagreement considering all possible transitions, all transitions from one category, or a specific transition from one category to another. In addition, it is possible to separate agreement between maps due to persistence from agreement due to correctly simulated change. This is important because in most applications the majority of cells stay the same during the study period, yielding a high rate of total agreement between observed and simulated maps which can misrepresent the ability of the model to correctly simulate the allocation of change (Pontius et al., 2004b; van Vliet et al., 2011). Pattern validation is commonly performed at multiple resolutions because comparison at the native resolution of the three maps fails to separate minor allocation disagreement, which refers to allocation disagreement at the native resolution that is counted as agreement at a coarser resolution, and major allocation disagreement, which refers to allocation disagreement at the native resolution and the coarse resolution (Pontius et al., 2011).

The *lulcc* package includes the first publicly available implementation of the pattern

validation method described in Pontius et al. (2011) (Robert Gilmore Pontius Jr, personal communication, 13 March 2016). In *lulcc*, three-dimensional contingency tables at multiple resolutions are represented by the ThreeMapComparison class. Two subclasses of ThreeMapComparison represent two types of information that can be extracted from the tables: AgreementBudget represents sources of agreement and disagreement between the three maps at several resolutions while FigureOfMerit represents figure of merit scores. The figure of merit, defined as the intersection of observed and simulated change divided by the union of these such that a score of one indicates perfect agreement and a score of zero indicates no agreement (Pontius et al., 2011), provides a useful summary of model performance. Plotting functions for ThreeMapComparison, AgreementBudget and FigureOfMerit objects allow the user to visualise model performance. In the following section of code the CLUE-S model output for Plum Island Ecosystems is validated by comparing observed land use/land cover maps for 1985 and 1999 and simulated map for 1999.

```
clues.tabs <- ThreeMapComparison(x=lu[[1]],
                                 x1=lu[[3]],
                                 y1=clues.result[[15]],
                                 factors=2^(1:8),
                                 categories=lu@categories,
                                 labels=lu@labels)

clues.agr  <- AgreementBudget(x=clues.tabs)
clues.fom  <- FigureOfMerit(x=clues.tabs)
```

The `factors` argument to the ThreeMapComparison constructor function shown above provides the aggregation factors, expressed as the number of cells in each direction, which control the various spatial resolutions at which pattern validation is performed. The above procedure was repeated for the ordered model output. The agreement budgets for the transition from Forest to Built for the two allocation procedures are shown in Figure 2.8, while Figure 2.9 shows the corresponding figure of merit scores.

2.3 Discussion

The Plum Island Ecosystems example highlights the key strengths of the new software package. One of the main advantages of *lulcc* compared to alternative software programs is that it performs the various stages of the modelling process shown in Figure 2.1 within the same environment. This improves workflow efficiency and reduces the likelihood of user errors because intermediate inputs and outputs exist in the same environment (Fiske and Chandler, 2011; Pebesma, 2012). In addition, it encourages interactive model building be-

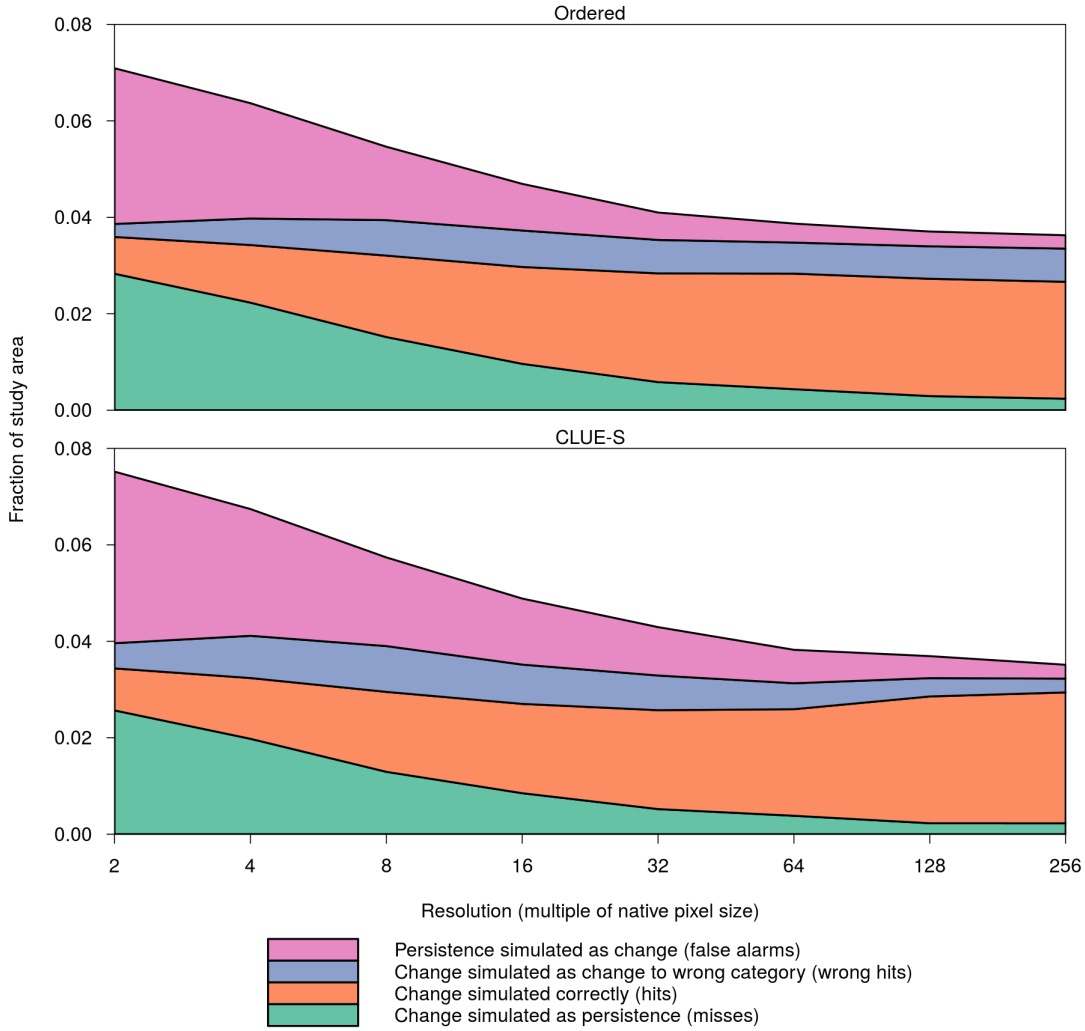


Figure 2.8.: Agreement budget for the transition from Forest to Built for the two model outputs considering reference maps at 1985 and 1999 and simulated map for 1999. The plot shows the amount of correctly allocated change increases as the map resolution coarsens.

cause seperate aspects of the procedure can easily be revisited. Furthermore, and perhaps most importantly, it improves the reproducibility of scientific results because the entire modelling workflow can be expressed programmatically and communicated as such with reasonable effort (Pebesma, 2012). The software exploits the openness of the R system which, through the package system, allows developers to contribute code, documentation and datasets in a standardised format to repositories such as the Comprehensive R Archive Network (CRAN) (Pebesma, 2012; Claes et al., 2014). As a result of this philosophy R users automatically have access to a wide range of sophisticated tools for statistical mod-

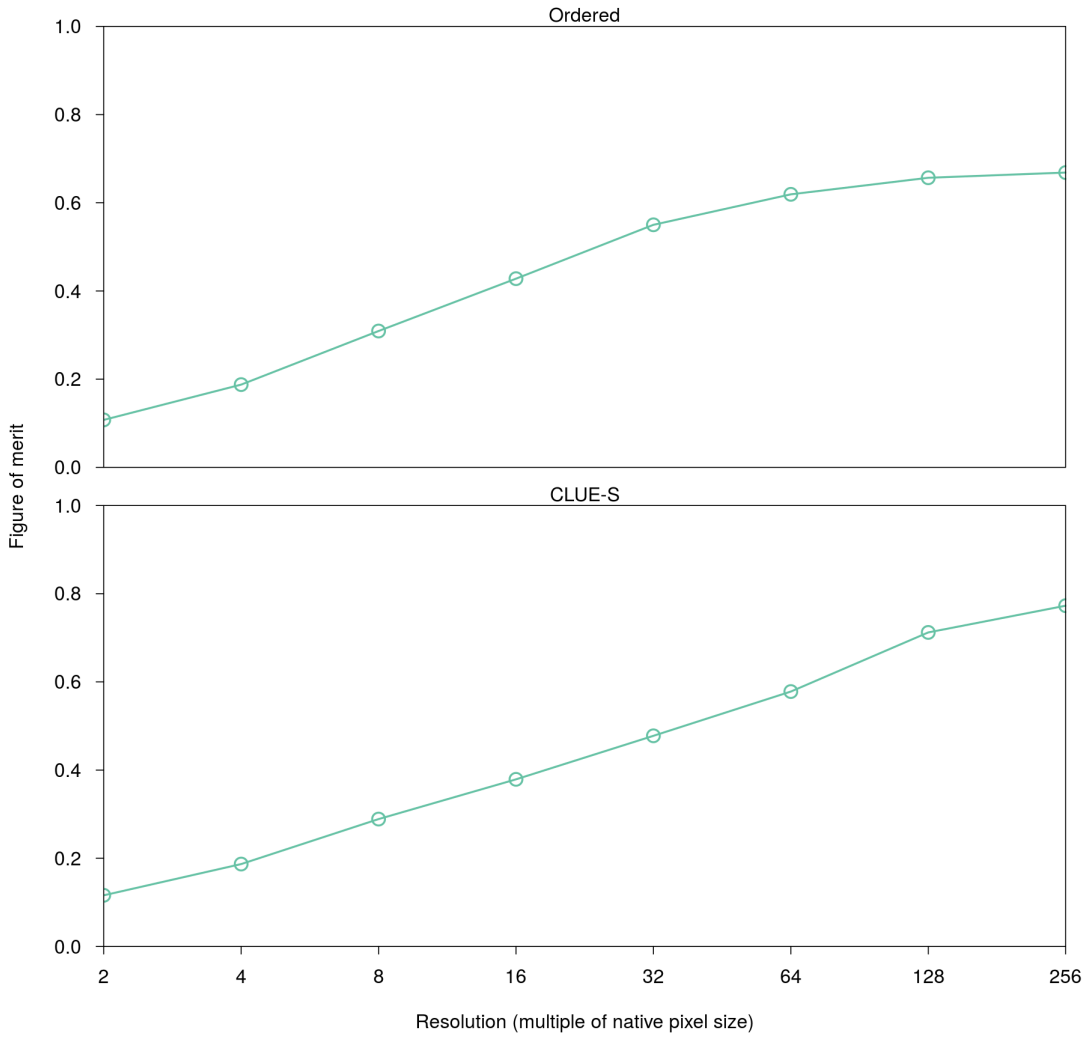


Figure 2.9.: Figure of merit scores corresponding to the agreement budgets depicted in Figure 2.8.

elling, data management, spatial analysis and visualisation. In *lulcc* this is particularly exploited through the use of *raster* for handling gridded spatial data and *randomForest* and *rpart* for statistical analysis. Furthermore, the package system ensures that *lulcc* will work across Windows, MacOS and Unix platforms, whereas many existing applications are platform dependent. The example also highlights the advantages of an object-oriented approach: land use change modelling involves several stages and without dedicated classes for the associated data it would be challenging to keep track of the intermediate model inputs and outputs.

The *lulcc* software package is substantially different from alternative environmental mod-

elling frameworks. It is specifically designed for land use change modelling whereas frameworks such as PCRaster and TerraME provide general tools that can be applied to various spatial analysis problems such as land use/land cover change, hydrology and ecology. Consequently, these tools are targeted towards the model developer rather than the end user. In contrast, most existing software programs for land use change modelling are designed for model users, with very few providing any way for users or developers to improve or even understand model implementations. One of the main goals of the *lulcc* package, therefore, was to reduce the gap between user and developer. The R system is well suited for this task, as Pebesma (2012) notes "the step from being a user to becoming a developer is small with R". One of the consequences of providing a modelling framework in R is that users of the software must become programmers (Chambers, 2000). This represents a different approach to the current practice of providing land use change software packages with graphical user interfaces (GUIs) and, consequently, could represent a steep learning curve for potential users unfamiliar with programming. To improve the accessibility of the software there is comprehensive documentation of the functions, classes and methods of *lulcc* (Appendix A), together with complete working examples which allow beginners to gain confidence using the software. Moreover, the fact that *lulcc* is free and open source software enables users to gain a deeper understanding of the software design and implementation. These factors mean that the package is well suited for use in an educational setting. Indeed, students at Clark University, Massachusetts have recently developed a video showing how *lulcc* can be used for pattern validation (Fitzgerald et al., 2016).

Despite its manifest advantages there remain some drawbacks to land use change modelling in R. Firstly, the lack of a spatio-temporal database backend to support larger datasets restricts the amount of data that can be used in a given application because R loads all data into memory (Gebbert and Pebesma, 2014). The *raster* package overcomes this limitation by storing raster files on disk and processing data in chunks (Hijmans, 2014). Where possible *lulcc* makes use of this facility; however, during allocation it is necessary to load the values of several maps into the R workspace at once because the allocation procedure must consider every cell eligible for change simultaneously. The generic `predict` function belonging to the *raster* package offers one possible solution to this problem, allowing predictive models to be used in a memory-safe way. In effect, this would mean that spatially explicit input data including observed land use/land cover maps and explanatory variables could be handled in chunks and only the resulting probability surface would have to be loaded into the R workspace. However, this is not currently implemented in *lulcc* because it is excessively time consuming compared to the current approach. Despite this limitation, since most applications involve a relatively small geographic extent or, in the case of regional studies (e.g. Verburg and Overmars, 2009; Fuchs et al., 2015), use a coarser map resolution, memory should not normally cause *lulcc* applications to fail. For example,

the CluesModel and OrderedModel objects from the above example each had a size of approximately 40Mb, which is easily handled by modern personal computers. On a 64-bit machine with Intel Core i3 @ 1.4 GHz and 4Gb RAM, the allocation methods for the two Model objects took 50 seconds and 8 seconds, respectively. The disparity between the runtime of the two models reflects the computational effort associated with the iterative procedure within the CLUE-S algorithm.

The software presented here is still in its infancy and there are several areas for improvement. The present allocation routines receive the quantity of land use/land cover change at each time point before the allocation procedure begins. However, some recent models do not impose the quantity of change but instead allow change to occur stochastically based on suitability. For example, StocModLcc (Rosa et al., 2013), a model designed for simulating tropical deforestation, deforests a cell if the probability of deforestation is less than a random number from a uniform distribution. The quantity of change is simply the number of cells deforested after each cell in the study region is considered for deforestation twice, with the probability of change, which depends on the allocation of previous deforestation events, updated after the first round. One advantage of this approach is that it accounts for uncertainty in the quantity and allocation of change simultaneously, whereas the current routines in *lulcc* only consider the allocation of change as a stochastic process. Other models such as LandSHIFT (Schaldach et al., 2011) receive demand at the national or regional level from integrated assessment models such as IMAGE (Stehfast et al., 2014) or Nexus Land-Use (Souty et al., 2012). Coupling *lulcc* with this class of model would be a valuable addition to the software because land use/land cover change is increasingly recognised as an issue with drivers and implications at local, regional, continental and global levels.

Allocation routines in *lulcc* are run by supplying an object inheriting from the Model class to the generic `allocate` function. However, after the first release of the software on CRAN some users criticised the software because it provided no way of allowing them to run models if the suitability maps had already been calculated elsewhere. The present version of *lulcc* retains the `allocate` function but exposes the allocation algorithms as plain R functions (`clues`, `ordered`, `clue`) which receive base R objects (e.g. vectors, arrays) and run for one timestep only. This allows users to implement the underlying allocation routines without introducing overhead which, in some cases, may be undesirable. One of the main strengths of *lulcc* is that multiple model structures can be explored within the same environment. Thus, the more allocation routines available in the package the more useful it becomes. Two existing land use change models, StocModLCC and SIMLANDER, are written in R and available as open source software. Future work could integrate these routines with *lulcc* to broaden the available model structures and, therefore, improve the

ability of *lulcc* to capture model structural uncertainty. The methods in the current version of *lulcc* only permit an inductive approach to land use change modelling. Deductive models are fundamentally different because they attempt to model explicitly the processes that drive land use/land cover change (Pérez-Vega et al., 2012). This means that, unlike inductive models, they can be used to establish causality between specific transitions and their driving factors (Overmars et al., 2007). Including this class of model in *lulcc* would allow inductive and deductive land use change models with different spatial resolutions to be dynamically coupled in order to better capture the complexity of the system (Moreira et al., 2009). The object-oriented design of `lulcc` is sufficiently flexible to support both inductive and deductive land use change models.

An important contribution of *lulcc* is to provide modules to assist with pattern validation; a crucial aspect of model development that is nevertheless frequently overlooked within the land use change modelling community (Rosa et al., 2014; van Vliet et al., 2016). A further improvement that could be made to the package is to incorporate more sophisticated ways of fitting and testing the predictive models that estimate land use/land cover suitability. For example, a routine to calculate the Total Operating Characteristic (TOC) (Pontius and Parmentier, 2014) would improve upon the ROC analysis currently provided. While ROC shows two ratios, hits/(hits+misses) and false alarms/(false alarms+correct rejections), at multiple thresholds, TOC reveals the quantities used to calculate these ratios, allowing greater interpretation of model diagnostic ability.

Free and open source software improves the reproducibility of scientific results and allows users to adapt and extend code for their own purposes and freely distribute changes to the wider community. Thus, members of the land use change modelling community are encouraged to participate in the future development of *lulcc*. Perhaps one of the simplest ways to improve the package is to experiment with the example datasets to identify bugs and areas for improvement. Those with more programming experience may wish to extend the functionality of the package themselves and contribute these changes upstream. In addition, existing land use change models can be included in the package by wrapping the original source code in R; a relatively straightforward task for commonly used compiled languages (C/C++, Fortran). Users may also develop their own R packages that depend on *lulcc* for some functionality: this is one of the strengths of the R package system. Finally, land use change modellers are invited to submit land use/land cover change datasets (observed and, if possible, modelled land use/land cover maps and spatially explicit explanatory variables) for inclusion in the package.

3 Modelling historical land use/land cover change in India

Over recent decades India has undergone substantial land use/land cover change as a result of population growth and economic development. This chapter reviews the state of the art in historical land use/land cover mapping and draws attention to the limitations of the currently available datasets. Subsequently, the *lulcc* implementation of the Change in Land Use and its Effects (CLUE) land use change model, driven by district-level inventory data, is used to generate land use/land cover maps for the Indian subcontinent for the period 1956–2010. Random forest models are used to describe the quantitative relationship between the present-day spatial distribution of land use/land cover, obtained from a state of the art contemporary product, and selected socio-economic and biophysical covariates. A post-processing algorithm is used to separate agricultural land use, which is treated as a homogeneous class during CLUE modelling, into cropland, fallow land and tree crops. Similarly, grassland is separated into grassland, shrubland and wetland because the latter two classes are not explicitly specified in the land use/land cover inventory data. In Chapter 5 the land use/land cover maps are supplied to a process-based land surface model to investigate the impact of large-scale environmental change on regional hydrology.

Material presented in this chapter is based on work described in Moulds, S., Buytaert, W. and Mijic, A. (2016). A spatio-temporal land use/land cover reconstruction for India (1956 – 2010). *Scientific Data*. In preparation. The land use/land cover change dataset, together with a set of scripts to implement the analysis, are available from the Figshare data repository (doi:10.6084/m9.figshare.3502802).

3.1 Literature review

India's population has risen dramatically in recent decades, from 361 million in 1951 to 1,221 million at the last national census in 2011 (James, 2011), driving substantial environmental change (Tian et al., 2014). Increasing population density and a changing economy have resulted in urban development and expansion (Gururaja and Sudhira, 2012), while India's green revolution, initiated in the mid-1960s to achieve food security for its growing population, has resulted in the transformation of natural vegetation and rainfed agriculture to intensively managed agricultural systems (Scott and Sharma, 2009; Smilovic et al., 2015). Land use/land cover change has important consequences for biodiversity and the sustainability of ecosystem services upon which individuals and communities depend (Turner et al., 2007). Cumulatively, it is a major driver of global and regional environmental change (Foley, 2005), influencing the surface energy and water balance (Feddema, 2005; Verburg et al., 2011) as well as global biogeochemical cycles (Houghton et al., 2012). Since the rate and magnitude of land use/land cover change varies over space and time (Tian et al., 2014), spatially explicit, historical datasets are necessary to accurately quantify the various environmental and societal impacts of change (e.g. Feddema, 2005; Verburg et al., 2011; Friedl et al., 2010; Foley et al., 2011; Meiyappan and Jain, 2012), support planning decisions and inform predictions about the quantity and location of future change.

Remote sensing provides one data source for mapping land use/land cover change (Friedl et al., 2002). In principle, satellite data are available for the last four decades (Houghton et al., 2012). However, the quality and availability of satellite images over much of the Earth's surface before the year 2000 is poor (Tian et al., 2014). The acquisition and storage of satellite images from Landsat instrument prior to 1999 was inconsistent and there are several gaps in the data records, especially during the 1980s when the Landsat programme was privatised (Houghton et al., 2012). Moreover, land use/land cover mapping in India is challenging because of pervasive cloud cover during the boreal summer (Ju and Roy, 2008; Tsarouchi et al., 2014). Finally, global and regional land use/land cover mapping using supervised and unsupervised methods require extensive ground truth data to extract meaningful classes from satellite imagery and to formally assess the accuracy of the resulting land use/land cover product (Loveland et al., 2000). Collecting sufficient data for these tasks, especially for assessing the accuracy of historical maps, is time consuming and expensive (Thenkabail et al., 2005). Existing efforts to document historical land use/land cover change at global and regional scales have used various spatial disaggregation algorithms to combine contemporary land use/land cover products derived from remote sensing imagery with national and sub-national inventory data (Meiyappan and Jain, 2012). Global datasets of historical land use/land cover change include the HYDE database (Klein Goldewijk et al., 2011) and the EarthStat global cropland and pasture dataset (Ramankutty

and Foley, 1999). These datasets have a 5 arcminute × 5 arcminute spatial resolution and cover the period 10,000BC – 2005AD and 1700–2007, respectively. In India, these global datasets rely on land use/land cover inventory data aggregated to state and national levels because of the difficulties in obtaining district-level data collected before 1998. As a result, they fail to capture the variability in the rate of change between districts, which can be considerable. For example, while the northern State of Uttar Pradesh shows a net increase in the area of agricultural land between 1956 and 2010, the area in 20% of district units has decreased.

Addressing this deficiency, Tian et al. (2014) developed an historical dataset for India which combines a contemporary land use/land cover dataset based on Resourcesat-1 satellite imagery (National Remote Sensing Agency, 2007) with a commercially available dataset of district-level land use/land cover inventory data (Indiastat, 2016). The dataset has a spatial resolution of 5 arcminute × 5 arcminute and covers the period 1880–2010. Central to the approach adopted to create this dataset, as well as the global datasets mentioned previously, is the assumption that the spatial distribution of historical land use/land cover is the same in relative terms as the present distribution (Tian et al., 2014). However, Meiyappan et al. (2014) has shown that this approach does not adequately reproduce historical change because it fails to account for changes in the relative importance of the various factors influencing land use/land cover change over time. Spatially explicit land use change models, which simulate future or historical land use/land cover change based on statistical analysis of the quantitative relationship between the contemporary land use/land cover distribution and various socio-economic and biophysical predictor variables (Verburg et al., 1999), have been suggested as an essential next step to improve historical land use/land cover reconstructions (Meiyappan et al., 2014; Tian et al., 2014). Previously, Ramankutty and Foley (1999) rejected this approach because of a lack of sufficient ancillary datasets at global and regional scales. However, in the last decade the availability of spatially explicit information for driving land use change models has increased dramatically. Recently, Fuchs et al. (2013) used a simple land use change model (on which the ordered routine in *lulcc*, described in the previous chapter, is based) to reconstruct a series of historical land use/land cover maps for Europe between 1950–2010. Meanwhile, Sohl et al. (2016) used the FORE-SCE land use change model (Sohl et al., 2007) to produce historical land use maps for the conterminous United States, while Tayyebi et al. (2015) used a land use change model based on two reference land use maps and historical agricultural and population census data to simulate the rate and allocation of historical and future land use change in the Ohio River Basin.

This chapter presents a new dataset showing historical land use/land cover change in India between 1956–2010. It uses the Change in Land Use and its Effects (CLUE) land

use change model (Veldkamp and Fresco, 1996a; Verburg et al., 1999), implemented in the *lulcc* software package introduced in Chapter 2, to combine district-level inventory data with a state of the art land use/land cover product for the year 2010 derived from Landsat data. The resulting dataset, which has a 5 arcminute × 5 arcminute spatial resolution and provides a map for each year of the study period, shows the evolution of agricultural land uses, forest, grassland, barren/sparsely vegetated land and urban land. Post-processing is used to separate agricultural uses into cropland, fallow land and tree crops, and grassland into grassland, shrubland and wetland. It is the first publicly available, spatially explicit dataset derived from district-level inventory data showing land use/land cover change in India over the second half the twentieth century. Furthermore, the methodology demonstrates the application of a dynamic land use change model to reconstruct historical land use/land cover change at regional scales.

3.2 Methodology

3.2.1 Input data

Administrative boundaries

Administrative boundaries were required to georeference district-level land use/land cover inventory data and subsequently to constrain CLUE model simulations. Contemporary district and state boundaries for India were obtained from the version 2.7 of the Global Administrative Areas Database (GADM, 2015). Since independence there have been several changes to Indian administrative boundaries. Therefore, contemporary GADM administrative boundaries were adjusted until they corresponded with boundaries that were consistent over the course of the study period. To identify the location of boundary changes a scanned, historical geological map showing district boundaries for 1971 was obtained from the European digital archive on soil maps (EuDASM) (Panagos et al., 2011).

Land use/land cover inventory data

The CLUE model is driven by a non-spatial demand scenario which specifies the total quantity of each land use/land cover category in the study region at each time point in the simulation. These quantities were derived from district-level land use/land cover inventory data. In India the collection of land use/land cover and agricultural data is decentralised and data collection methods vary between the various States and Union Territories (territories administered by the Central Government). These can be separated into three main groups:

1. States and Union Territories which have been cadastrally surveyed (i.e. property boundaries within the administrative unit have been mapped). In this case village revenue agencies survey all fields during each growing season to compile crop and irrigation inventory data;

2. Kerala, Orissa and West Bengal. Villages in these States do not have their own revenue agency and statistics are collected from a random sample of 20% of villages, such that over a five year period each village is surveyed once, and extrapolated to all villages;

3. States and Union Territories without a formal reporting system. In this case inventory data are based on estimates made at the village level.

In all cases the raw village level data are aggregated to district and State administrative levels. Figure 3.1 shows that the majority of district units in the study region are subject to a full survey based on cadastral survey information. The Directorate of Economics and Statistics, which compiles inventory data at the national level, does not make district-level data collected before 1998 freely available. Instead, data for the entire study period, except the period 1993–1997 which, for unknown reasons, was not available, was purchased from Indiastat (Indiastat, 2016). An additional source of district-level survey data for 19 States and Union Territories for the period 1966 to 2009 was obtained from the Icrisat Village Dynamics in South Asia (VDSA) project (Icrisat, 2016). This data originates from the same source as the Indiastat data but, as part of the VDSA project, has already undergone some data quality control and additionally includes data for the period 1993–1997. It was therefore suitable for infilling some missing or anomalous data points in the Indiastat dataset.

Initial land use/land cover

A contemporary land use/land cover map was used to perform regression analysis and supply the intitial condition for the CLUE model. The map was obtained from GlobeLand30 (Chen et al., 2015), a state of the art global land use/land cover product for 2010 developed using 30m Landsat data. The high spatial resolution of this dataset overcomes many of the drawbacks associated with coarser land use/land cover products such as MODIS-IGBP, MODIS-UMD (Friedl et al., 2010) and GlobeCover (Arino et al., 2008) products, resulting in an overall classification accuracy of approximately 80% (Chen et al., 2015). One potential discrepancy between the GlobeLand30 dataset and the land use/land cover inventory data arises from the possibility that fallow land, considered an agricultural land use, is interpreted by the classification algorithm as grassland, shrubland or bareland according to its physical properties. However, the GlobeLand30 methodology employs a sophisticated

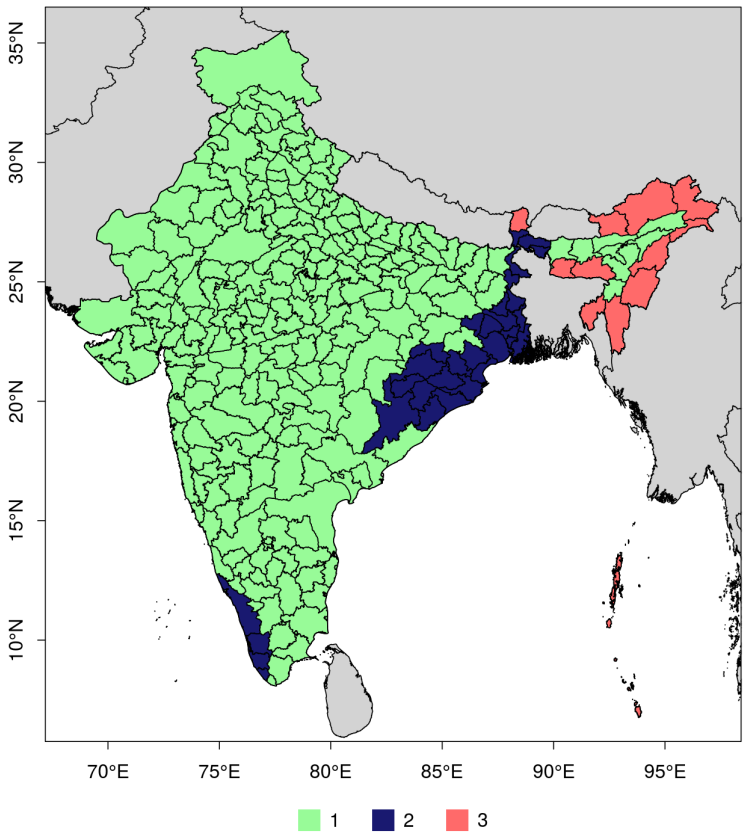


Figure 3.1.: Survey group to which each district unit in the study region belongs. The majority of districts belong to survey group 1, shown in green, which are subject to a full survey based on cadastral survey information. In Kerala, Orissa and West Bengal, shown in blue, agricultural data is collected from a random sample of 20% of districts in any given year such that in a five year period all districts will have been surveyed once. Lastly, certain districts in north-east India, shown in red, do not have a formal reporting procedure and inventory data is based on estimates made at the village level.

classification procedure which, in addition to pixel-based classification, employs object-based classification and knowledge-based verification to ensure fallow land is classified as cultivated land (Chen et al., 2015).

Biophysical and socio-economic predictor variables

The CLUE model is based on statistical relationships between the spatial distribution of contemporary land use/land cover and selected spatially explicit biophysical and socio-economic covariates. Seven soil quality maps, representing nutrient availability, nutrient retention capacity, rooting conditions, oxygen availability to roots, excess salts, toxic-

ity and workability, were obtained from the Harmonised World Soil Database (HWSD; FAO/IIASA/ISRIC/ISS-CAS/JRC, 2012). Maps showing elevation, slope and aspect, ancillary datasets to the HWSD methodology, were obtained from the same source. These maps were derived from the NASA Shuttle Radar Topographic Mission digital elevation model with a 3 arcsecond × 3 arcsecond spatial resolution before aggregation to 5 arcminute × 5 arcminute grid cells. Elevation at the coarse resolution is the median elevation considering grid cells at the fine resolution. Fine resolution slope maps are used to create seven coarse resolution maps where the value of each cell represents the percentage of fine grid cells belonging to one of seven slope classes. Aspect was similarly divided into five classes representing north, east, south, west and undefined (slopes below 2%) aspect.

Population density maps for 1950, 1960, 1970, 1980, 1990, 2000 and 2005 at 5 arcminute × 5 arcminute spatial resolution were obtained from the HYDE database (Klein Goldewijk et al., 2010). These maps were produced by disaggregating subnational population census data to 5 arcminute × 5 arcminute grid cells based on likelihood coefficients derived from spatially explicit biophysical and socio-economic variables such as distance to roads, slope and night-time lights (Klein Goldewijk et al., 2010). The NDVI3g dataset (Pinzon and Tucker, 2014), derived from NASA's Advanced Very High Resolution Radiometer (AVHRR) instrument, provides a bi-monthly time series of normalized difference vegetation index (NDVI) between 1981–2013 at 5 arcminute × 5 arcminute resolution. Since vegetation indices are proxies for photosynthetic activity, the temporal evolution of NDVI over the course of a growing season can be used to differentiate land use/land cover categories (Thenkabail et al., 2005). While the NDVI3g dataset is too coarse for classifying individual categories, it still contains relevant information about the spatial distribution of land use/land cover which can be effectively assimilated through statistical analysis.

3.2.2 Data preparation

The scanned geological map showing district boundaries for 1971 was geolocated and used together with historical records to identify changes in administrative boundaries over the course of the study period. Contemporary district polygons from GADM were manually dissolved until the resulting area corresponded with boundaries shown by the historical map. As a result of the changes the mean district unit area increased from 5307 km^2 ($n = 594$, $\sigma = 4673\ km^2$) in the original GADM map to 10650 km^2 ($n = 296$, $\sigma = 8777\ km^2$) in the modified version.

The relevant tiles of the contemporary land use/land cover dataset were downloaded from the GlobeLand30 Web service, reprojected from Universal Transverse Mercator to Albers Equal Area projection and mosaicked together into a single raster dataset with a

30m × 30m spatial resolution. The resulting image was reclassified to seven land use/land cover types, as shown in Table 3.1. From this map a separate binary map for each land use/land cover type was created where 1 indicates presence and 0 indicates absence. These maps were resampled to a spatial resolution of approximately 9km × 9km (equivalent to 5 arcminute × 5 arcminute in latitude/longitude) taking the sum of the fine resolution cells so that the resulting coarse images displayed the fraction of each grid cell belonging to the respective land use/land cover types. Biophysical and socio-economic predictor variables were resampled to the same projection and spatial resolution as the fractional land use/land cover maps. For each year covered by the NDVI3g dataset, the raw images were simplified by taking the mean, minimum, maximum and range during the entire year and during the *Kharif* (monsoon; June–October) and *Rabi* (dry; November–March) growing seasons. Lastly, the modified administrative area map was reprojected to Albers Equal Area projection and each polygon was converted to a raster image with the same spatial resolution as the other spatial data. Cell values in these images represent the fraction of the cell belonging to the corresponding district unit, such that cells entirely contained in the polygon had a value of one while cells on district boundaries had values between zero and one.

Inventory data obtained from Icrisat and Indiastat were homogenized by checking for inconsistencies, identifying different spellings of administrative unit names, and applying consistent formatting. Although many districts had data for only part of the study period, either because the data was missing or the district was dissolved or created during the study period as a result of boundary changes, all district data were mapped to a common annual time series between 1956–2010. The resulting data files were aggregated to correspond with the district units of the modified GADM administrative area map. This yielded two new data files in which district units are associated with a unique time series of land use/land cover data from Indiastat and Icrisat, respectively.

The inventory data is based on nine land use/land cover categories: (a) Forests, (b) area under non-agricultural uses, (c) barren and unculturable land, (d) permanent pastures and other grazing land, (e) miscellaneous tree crops, (f) culturable waste land, (g) fallow land other than current fallows, (h) current fallows, and (i) net area sown. This classification scheme was simplified to provide the demand scenario input to the CLUE model (Table 3.1). Following Tian et al. (2014), the change in urban area was estimated by multiplying the time series of area under non-agricultural uses, which includes artificial surfaces, by a scale factor equal to the area of artificial surfaces from GlobeLand30 divided by the area under non-agricultural uses for 2010. The area of water and permanent snow and ice were also obtained from the GlobeLand30 map and, making the assumption that they remain constant during the study period, subtracted from the area under non-agricultural

Table 3.1.: Translation of land use/land cover classification schemes of inventory data and GlobeLand30 product to the classes used in the CLUE modelling procedure.

CLUE modelling	Inventory class	GlobeLand30 class
Forest	Forest	Forest
Urban	Area under non-agricultural uses	Artificial surfaces
Agricultural uses	Net area sown Miscellaneous tree crops Current fallows Fallow land other than current fallows	Cultivated land
Grassland	Permanent pasture Culturable wasteland	Grassland Shrubland Wetland
Barren/sparsely vegetated land	Barren and unculturable land	Tundra Bareland
Water	Area under non-agricultural uses	Water bodies
Permanent snow and ice	Barren and unculturable land	Permanent snow and ice

uses. The remaining area under non-agricultural uses was allocated to barren/sparsely vegetated land.

District-level time series data contained various types of errors. Three main problems were encountered:

1. Data points labelled as one land use/land cover class but belonging to another;
2. Data points labelled with the wrong area units;
3. Missing data points and outliers.

Data quality control was performed at the district level by plotting the time series of each land use/land cover class, identifying quality issues and infilling or correcting problems as appropriate. Misclassified data points were reclassified to the correct land use/land cover class while data points labelled with the wrong units were multiplied by an appropriate conversion factor. Missing data points and outliers were infilled using linear interpolation or, if the problem occurred in the first or last data points, last observation carried forwards. Data smoothing using locally weighted least squares regression (LOESS) was performed because the implementation of the CLUE model is sensitive to small deviations in the direction of change of land use/land cover categories.

The final processing step was to calibrate the district-level time series data with land use/land cover quantities from the GlobeLand30 dataset, following Tian et al. (2014). This step was necessary because the CLUE model is sensitive to the initial condition and usually fails to converge on subsequent time points if the demand scenario differs markedly from the initial quantities. In each district unit a calibration factor was determined for each category by dividing the quantity from GlobeLand30 with the corresponding quantity for 2010 from the inventory data. The time series of each category was multiplied by the calibration factor. Differences between the district area and the sum of the various categories at each time point were resolved by adjusting the area of grassland and barren/sparsely vegetated land.

3.2.3 Land use change modelling procedure

Statistical analysis and the spatial allocation of land use/land cover change was performed in the R environment using the *lulcc* software package developed in Chapter 2. Random forest, a non-parametric ensemble learning method that is robust against overfitting and efficiently handles datasets with a large number of input variables (Breiman, 2001; Svetnik et al., 2003), was used to fit models describing the spatial distribution of land use/land

cover, derived from the GlobeLand30 dataset, with selected biophysical and socio-economic covariates (soil quality, topography, population density, NDVI). The R package *randomForest*, which provides a convenient interface to the original Fortran code of Breiman (2001), was used for statistical analysis. The random forest algorithm constructs multiple classification trees using different bootstrap samples of the data (Breiman, 2001), splitting tree nodes using the best split among a random selection of variables (Breiman, 2001; Liaw and Wiener, 2002). For regression tasks the results from individual trees are combined by taking the mean prediction. The input dataset, comprising 45,125 data points (equal to the number of cells at 5 arcminute resolution covering the Indian subcontinent), was split into training and test sets according to a 70:30 ratio. All models (crop, forest, grass, barren/sparsely vegetated land, urban) were trained and tested on the same data partitions.

The CLUE model, constrained with demand scenarios constructed from the land use/land cover inventory data, was applied to each district in turn. Five land use/land cover categories were modelled: forest, cropland, grassland, urban and barren/sparsely vegetated land. Water and permanent snow and ice classes were also included in the simulation but assumed to be constant over the study period. To account for cells located over administrative boundaries, which may have partial membership to two or more districts or be situated on the coast, the simulation included an eighth class representing the fraction of each cell in the simulation that does not belong to the district in question, such that cells entirely contained in the district have a fill value of zero while boundary cells have a value between zero and one. This was necessary to fulfill the requirement of CLUE that land use/land cover fractions in each cell sum to one. During the simulation, which proceeded backwards in time from 2010, dynamic predictor variables (population density, NDVI) were updated at each time point. During 1956–1981 when NDVI was not available the NDVI predictor variables for 1982 were used. The resulting district-level land use/land cover maps were combined to generate an historical land use/land cover change dataset for the entire Indian subcontinent.

The CLUE model output was post-processed in order to divide total agricultural uses into cropland, fallow land and tree crops, and grassland into grassland and shrubland. In the former case the fraction of each grid cell belonging to agricultural land uses, considering each district unit in turn, was multiplied by the district-wide fraction of total agricultural uses devoted to cropland, fallow land and tree crops, respectively, according to the district-level land use/land cover inventory data. To separate grassland and shrubland classes, which are grouped together in the land use/land cover inventory data and, consequently, the CLUE model output, the proportion of shrubland and grassland in each coarse grid square from 2010, derived from the the GlobeLand30 land use/land cover dataset, was assumed to hold throughout the study period.

Table 3.2.: Out-of-bag prediction error and coefficient of determination for the random forest models of the spatial distribution of forest, urban, agricultural uses, grassland and barren/sparsely vegetated land

	OOB	R^2 (test set)
Forest	0.89	0.89
Urban	0.65	0.65
Agricultural uses	0.88	0.88
Grassland	0.57	0.58
Barren/sparsely vegetated land	0.88	0.88

3.3 Results

3.3.1 Statistical analysis

In random forests a given tree is constructed using a bootstrap sample from the training set containing about two-thirds of cases, with the remaining cases used to calculate the prediction error of the tree. The out-of-bag (OOB) prediction error, calculated internally by the model fitting algorithm, is the average prediction error across all trees (Breiman, 2001). As a sanity check, the coefficient of determination (R^2) of each random forest model was computed from the test set. Table 3.2 shows the respective scores for each random forest model, demonstrating that models of crop, forest and barren/sparsely vegetated land have good performance while models of grass and urban have adequate performance.

3.3.2 Spatial allocation

Pattern validation of the CLUE model output was not possible because there are no reliable land use/land cover maps for India before the year 2000. Instead, the CLUE model output was evaluated by comparing the allocated area of each land use/land cover class with the corresponding demand scenario derived from district-level inventory data. Differences between the specified area of each land use/land cover class and the area actually allocated arise when the CLUE model fails to converge at a specific time point. Figures 3.2 and 3.3 show the results of the comparison aggregated to the national level, showing good agreement between the calibrated inventory data and the modelled area. The figures also show the area of the respective land use/land cover types from the raw inventory data and after data quality control was performed. The data repository includes plots showing the results of this analysis at state and district levels. Figure 3.4 shows the spatial distribution of land use/land cover in 2010 derived from GlobeLand30. These maps were used to fit the random forest models of the respective land use/land cover classes and provided the initial

condition for the CLUE model simulation. Figure 3.5 shows the location of land use/land cover change between 1960 and 2010. Lastly, Figure 3.6 shows the spatial distribution of fallow land between 1999–2004 in order to highlight the impact of the drought in July 2002 on agricultural production.

3.4 Discussion

Historical land use/land cover maps are necessary to understand interactions between humans and the environment and to develop policy concerning agricultural development, urban expansion and the protection of ecosystem services. They are an essential input to assessments of global and regional change because land cover influences fluxes of energy, momentum and water fluxes between the land and atmosphere as well as biogeochemical cycles. The study highlights the use of land use change modelling to reconstruct historical land use/land cover change at the regional level using various publicly available global and regional datasets. This approach improves upon previous methods to reconstruct historical land use/land cover change in India which are based on the assumption that the spatial distribution of historical land use/land cover is the same as the present distribution (e.g. Tian et al., 2014) and, as a result, fail to account for temporal variability in the underlying factors which influence the rate and location of change. By contrast, the CLUE land use change model employed in the current study dynamically simulates land use/land cover change in India considering various socio-economic and biophysical drivers of change. Moreover, the methodology enabled the assimilation of available satellite data by including the AVHRR NDVI3g dataset as a covariate in the random forest models of vegetated land use/land cover categories. Figure 3.2 shows that the dominant transition during the study period was from grass to agricultural uses, which is consistent with the findings of previous studies of land use/land cover change in India (Tian et al., 2014). As Figure 3.5 shows, this conversion has occurred throughout the country, especially in central regions such as the state of Madhya Pradesh. Furthermore, in the north-west state of Rajasthan a large area of land previously devoted to grass and barren/sparsely vegetated land has been converted to agricultural uses. Urbanisation has occurred in every part of the country with the most noticeable change in New Delhi, India's capital city. These results are broadly consistent with sub-national studies of land use/land cover change (e.g. Tsarouchi et al., 2014). The new dataset could be used for hydrological and earth system modelling to assess the impact of land use/land cover change on earth system functioning. In addition, it may be useful in a policy context to identify "hot-spots" of land use/land cover change in India and inform planning decisions about future development.

The land use/land cover inventory data is subject to considerable uncertainty from var-

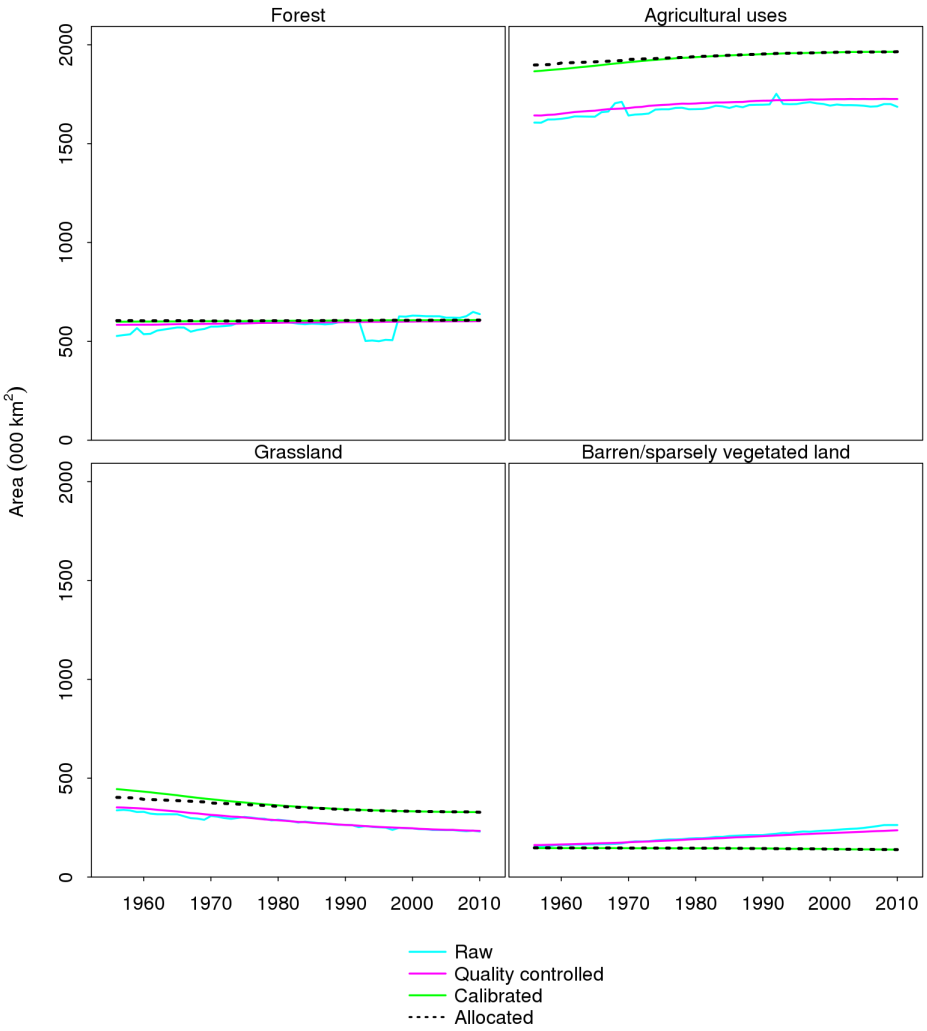


Figure 3.2.: Comparison of the area of crop, forest, grass and barren/sparsely vegetated land use/land cover classes at the national level considering the raw inventory data, inventory data after quality control and calibration with GlobeLand30 product, and area actually allocated. Differences between the specified area (green line) and the area actually allocated (dashed line) arise when the CLUE model fails to converge on a solution before the maximum number of iterations allowed is reached. The area of calibrated area of Barren/sparsely vegetated land appears to diverge from the quality controlled area because of the way in which calibration is carried out, whereby the entire time series is multiplied by a factor obtained by dividing the 2010 value from inventory data with the corresponding value from the GlobeLand30 product.

ious sources. Two types of uncertainty must be considered: uncertainty in the quantity of land change at the district level and uncertainty in the spatial allocation of land use change

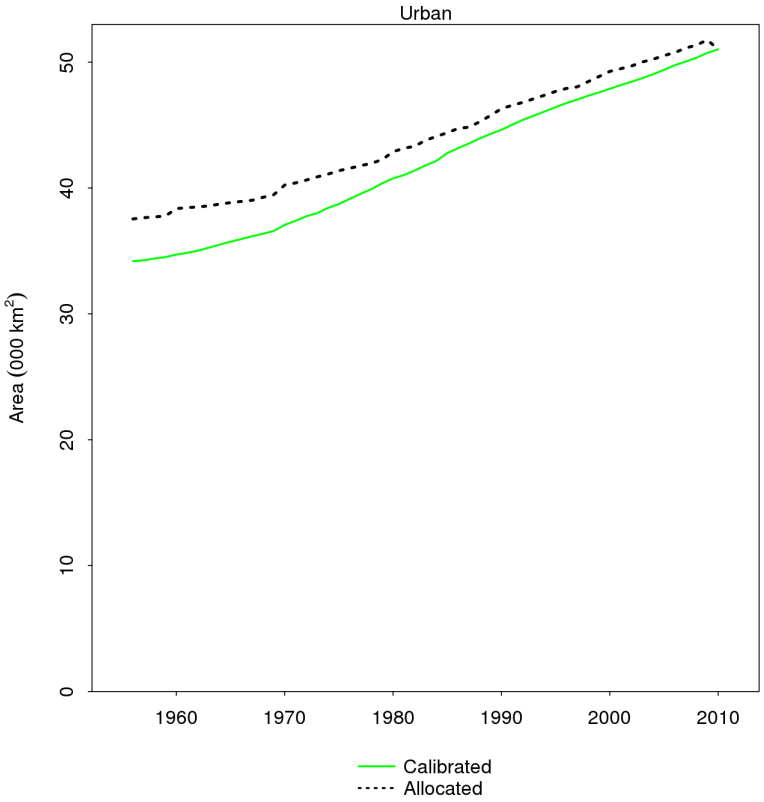


Figure 3.3.: Comparison of the area of urban land use at the national level considering the inventory data after calibration with GlobeLand30 product and the area actually allocated.

within each district unit. Raw district-level land use/land cover inventory data, obtained from Indiastat and Icrisat, contained a large amount of noise which had to be manually removed before supplying it to the CLUE model. Identifying between natural variability and erroroneous data points in the inventory data is a challenging task and it is highly likely that some errors have been interpreted as natural variability and vice versa. To ensure decisions made during data quality control are open, reproducible and amendable the scripts used to process the data are included alongside the land use/land cover change dataset together with instructions about how they can be modified. Uncertainty also arises from the procedure to create a consistent set of district boundaries for the study period by manually dissolving contemporary GADM boundaries. It is likely that during the procedure, in which boundary changes were identified using a scanned and geolocated historical map, some minor boundary changes were not identified. It is also possible that there have been differences between the perception of administrative boundaries amongst those in charge of collecting village-level statistics and those responsible for aggregating them

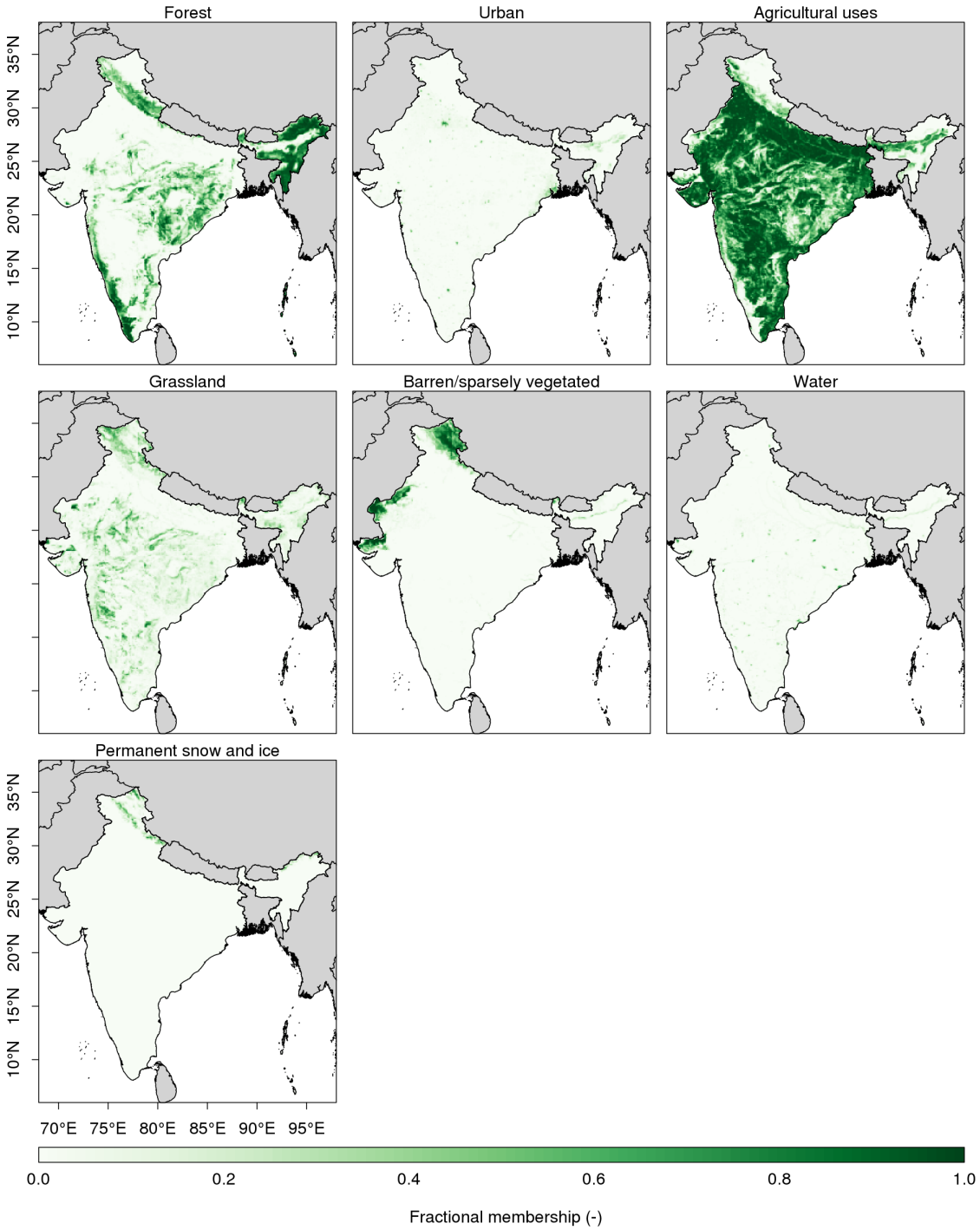


Figure 3.4.: Land use/land cover in 2010, derived from the GlobeLand30 dataset (Chen et al., 2015).

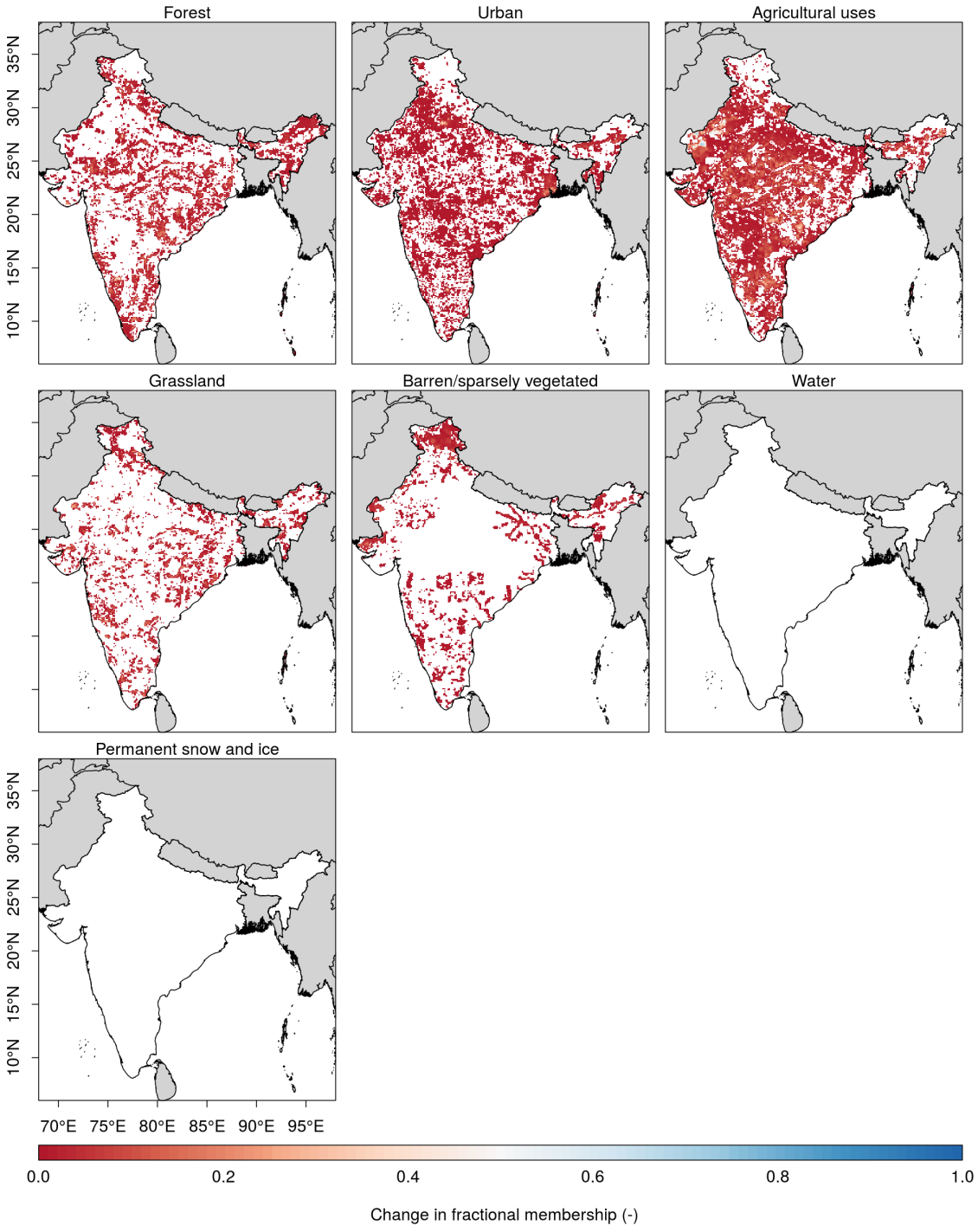


Figure 3.5.: Location of change of crop, forest, grass and barren/sparsely vegetated land use/land cover classes between 1960–2010.

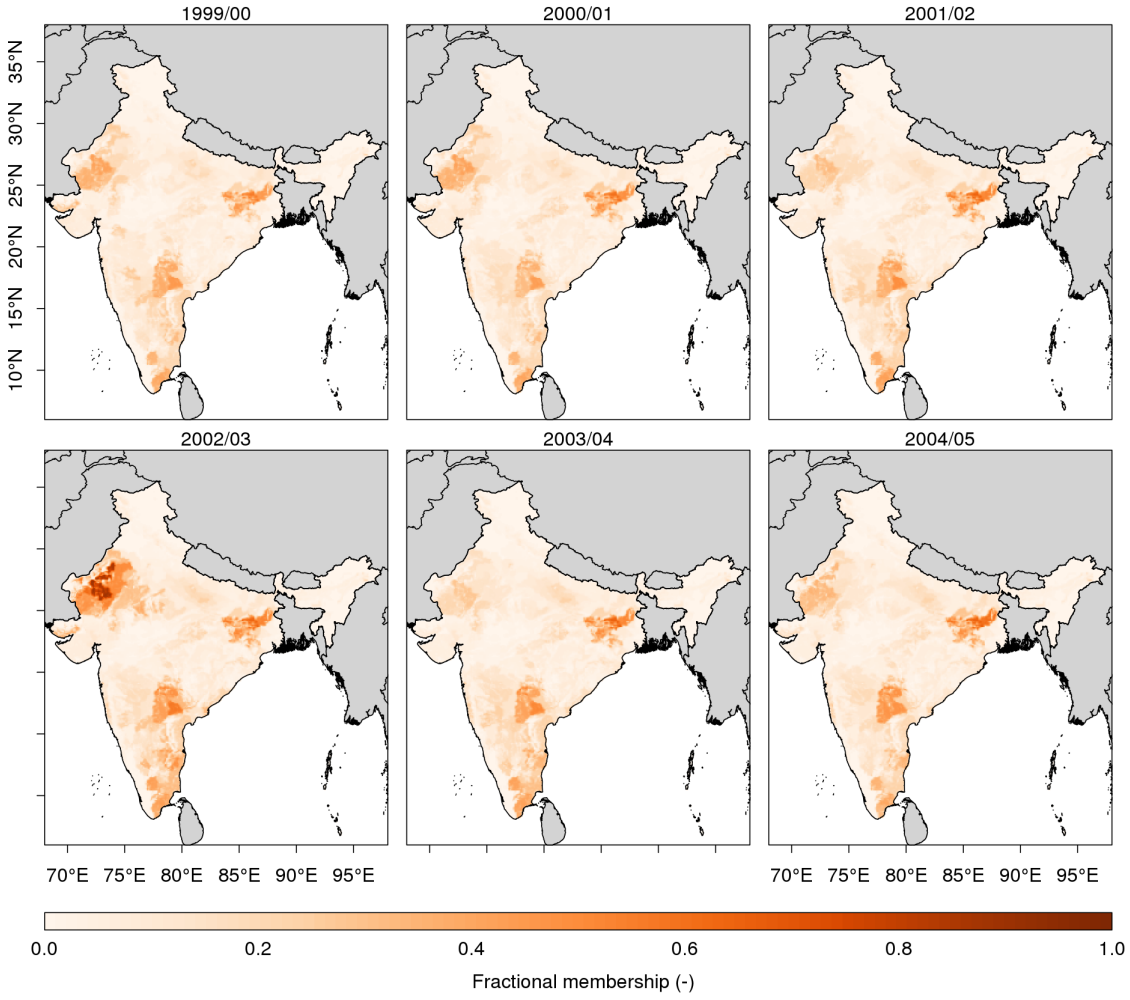


Figure 3.6.: Spatial distribution of fallow land between 1999–2004, highlighting the impact of the July 2002 drought on cropland extent, especially in parts of north-west India.

to district and state levels. For these reasons it is likely that to some extent the boundaries used in the present study do not accurately reflect the boundaries used on the ground.

At the level of individual grid cells there is uncertainty about the information provided by the spatial input data. In particular, while the GlobeLand30 dataset represents the state of the art in global land use/land cover mapping, it is nevertheless associated with uncertainty in both the quantity and spatial distribution of land use/land cover in 2010 (Chen et al., 2015). Figure 3.2 shows there is a large discrepancy between the total quantity of each land use/land cover type in the inventory data and the equivalent land use/land cover

quantity in the GlobeLand30 dataset, with agricultural uses in 2010 around 20% higher at the national level compared to the raw inventory data. This likely reflects an underestimation of agricultural uses in the inventory data and an overestimation of agricultural uses by the remote sensing dataset. One way to account for this uncertainty would be to repeat the analysis using alternative remote sensing datasets. The Harmonised World Soil Database, which draws together many national and regional datasets, represents the state of the art in global soil information (FAO/IIASA/ISRIC/ISS-CAS/JRC, 2012). However, it is associated with considerable uncertainty because of the scarcity of soil profile observations (Hengl et al., 2014). Notwithstanding these shortcomings, Figure 3.2 shows there is good agreement between the raw, quality assessed and calibrated inventory data in terms of the trajectory of change during the study period. It was not possible to validate the spatial allocation of change due to the lack of observed land use/land cover datasets for India before the year 2000. It would in principle be possible to compare simulated maps between 2000-2010 with various remote sensing products. However, since there has been a relatively small amount of land change during the period, the comparison would mainly serve to demonstrate differences between GlobeLand30 and alternative land use/land cover products rather than to validate the model outputs.

The CLUE model is strongly influenced by the statistical analysis of the spatial distribution of contemporary land use/land cover and its covariates. The random forest models employed in the present study use several publicly available datasets including soil quality, topography, population distribution and, for the period 1982–2010, NDVI, in addition to the GlobeLand30 dataset from which the dependent variable of each model is derived. Recognising that other members of the community may have access to alternative or additional datasets outside the public domain, the scripts used in the analysis include clear instructions about incorporating additional data into the statistical analysis. Table 3.2 shows the performance of the random forest model for the spatial distribution of grassland is the worst performing model. This can be explained by the fact that grassland thrives under a wide range of biophysical conditions and appears in the absence of agricultural and urban areas (Verburg and Overmars, 2009). As a result, the spatial distribution of grassland may be influenced more by the low suitability of land to agriculture and urban areas than the suitability of grassland itself. It is likely that the poor performance of the statistical model for grass has contributed to the failure, in some districts, of the CLUE allocation routine to converge on a solution, resulting in the discrepancy between the calibrated inventory data and the area actually allocated shown in Figure 3.2. One solution to this problem would be to use an alternative spatial allocation model, such as the ordered procedure included in *lulcc*, which does not simulate competition between land use/land cover categories and therefore yields an exact solution. The main barrier to this approach is the lack of biophysical and socioeconomic data at an appropriate spatial resolution with

which to fit the predictive models upon which the allocation procedure depends. In the future, however, this approach may become more feasible.

Defining the spatial distribution of land use/land cover is a necessary first step towards developing datasets of land management and land use intensity. According to Houghton et al. (2012), the lack of such datasets is one of the main reasons for the large uncertainty about the influence of the biosphere on global and regional change. Sacks et al. (2009) argues that land management activities can be as important or more important than land use/land cover dynamics for understanding the influence of the land surface on climate and, therefore, accurately simulating past and future climate. In India, while the area devoted to agriculture has increased by approximately 5% during the study period, the fraction of cropland that is irrigated at least once during the agricultural year has increased by about 60% according to agricultural inventory data. There has also been a dramatic rise in the amount of fertiliser applied to agricultural land. As Biemans et al. (2013) observes, since most land suitable for agricultural activities is already in use future gains in productivity must be achieved through agricultural intensificiation. Thus, understanding the spatial characteristics of land management practices and land use intensity is necessary to improve predictions of future environmental change and support planning decisions.

3.5 Chapter summary

Over recent decades there has been substantial land use change in India as a result of population growth and economic development. Since the rate and magnitude of change varies over space and time, spatially explicit datasets are necessary to accurately quantify the environmental and societal effects of land use change as well as to inform predictions about the quantity and location of future change. Current datasets for India, including global and regional products, are based on the assumption that the relative spatial distribution of land use/land cover remains constant over time. Moreover, global datasets are based on national or state-level inventory data which fail to adequately capture the variability in the rate of change between districts. This chapter applied the CLUE regional land use change model, driven by district-level land use/land cover inventory data, to reconstruct historical land use/land cover change in India between 1956–2010. Random forest models were used to describe the relationship between the spatial distribution of contemporary land use/land cover and selected biophysical and socio-economic covariates. The resulting dataset has a 5 arcminute $\times$ 5 arcminute spatial resolution and shows the fractional membership of each grid cell in the study region to forest, urban, agricultural uses, grassland, barren/sparsely vegetated land, water and permananent snow and ice on an annual basis during the study period. Post-processing was used to separate agricultural uses into crop-

land, fallow land and tree crops and grassland into grassland, shrubland and wetland. The dataset is the first to provide annual maps of land use/land cover for India over the second half of the twentieth century and, furthermore, represents a proof-of-concept for using a land use change model for simulating historical land use/land cover change at the regional level.

4 Reconstructing historical changes in India's irrigated area

India's green revolution has resulted in large-scale agricultural intensification to feed its growing population. Irrigation has been central to this transformation, providing a buffer against the variability of the South Asian monsoon and allowing farmers to grow crops in the dry winter season. Historical datasets showing the change in irrigated area over time are necessary for assessments of global and regional change and provide insight into human-environment interactions. This chapter reviews the state of the art in irrigated area mapping and highlights the weaknesses of existing approaches as well as areas for improvement. Based on the issues identified an improved methodology is proposed for India. District-level agricultural inventory data about the irrigated area of 25 crop types is spatially disaggregated using historical maps of cropland extent, developed in Chapter 3, and maps of biophysical crop suitability. The resulting dataset has a 5 arcminute $\times$ 5 arcminute spatial resolution and provides a map for each crop type on an annual basis. The maps show good agreement with existing contemporary maps of irrigated area; however, there are no equivalent datasets of irrigated area before the year 2000 available for comparison. The dataset is associated with considerable uncertainty at the grid cell scale arising from the district-level inventory data as well as the spatial disaggregation procedure. Consequently it is advisable to use the dataset for global and regional scale applications rather than local-scale studies that are sensitive to the values of individual grid cells.

Material presented in this chapter is based on work described in Moulds, S., Buytaert, W. and Mijic, A. (2016). Mapping India's green revolution: a spatio-temporal dataset of irrigated land, 1956 – 2010. *Hydrology and Earth System Sciences*. In preparation. As in the previous chapter the irrigated area dataset, together with a set of scripts to implement the analysis, is available from the Figshare data repository (doi:10.6084/m9.figshare.3504092).

4.1 Literature review

India's green revolution, initiated in the mid-1960s to achieve food security for the nation's rapidly growing population (Singh, 2000), has driven large scale land use/land cover change from natural land cover and low-intensity cropland to high-input agricultural systems (Evenson and Gollin, 2003; Foley et al., 2011). The widespread practice of irrigation, which provides a buffer against intraseasonal monsoon variability and allows farmers to grow crops throughout the year, has been essential to this transformation (Alauddin and Quiggin, 2008; Shukla et al., 2014). While the green revolution has enabled the country to become self sufficient in food production and supported the livelihoods of millions (Wada et al., 2013), it has also caused a dramatic increase in the exploitation of water resources; particularly the vast aquifers of the fertile Indo-Gangetic plains in northern India where groundwater levels in places with intensive irrigation have lowered considerably (Rodell et al., 2009; Gleeson et al., 2012). The pressure on water resources will intensify as the country experiences major social and economic change (Zhao et al., 2006; James, 2011; Elliott et al., 2014). Furthermore, the South Asian monsoon, which provides up to 85% of the total annual rainfall across the Indian subcontinent (Turner and Annamalai, 2012), shows increasing variability as a result of climate change and other factors (Bollasina, 2014; Singh et al., 2014). Given these threats to food security, irrigation will continue to be an essential feature of Indian agriculture (Biemans et al., 2013; Smilovic et al., 2015); indeed, irrigated area is projected to increase in the future (Neumann et al., 2011; Elliott et al., 2014; Haddeland et al., 2014).

Spatio-temporal datasets showing the change in extent and distribution of irrigated area over time are critically important for sustainable water resources management (Wisser et al., 2008; Nazemi and Wheater, 2015a; Siebert et al., 2015). They provide insight into the drivers of agricultural intensification, highlight areas of unsustainable exploitation of water resources and improve the projections of future irrigation requirements to ensure food security (Döll and Siebert, 2002; Wisser et al., 2008). Irrigated area datasets are also recognised as an essential input to many assessments of global and regional change (e.g. Pokhrel et al., 2012; Nazemi and Wheater, 2015a). As Houghton et al. (2012) points out, the lack of historical, spatially explicit data showing the change in land use intensity is a major cause of uncertainty about the effect of the biosphere on large-scale environmental change. This is particularly the case in India where there is growing evidence that irrigation over the last 60 years has influenced the behaviour of the South Asian monsoon (Douglas et al., 2006; Sen Roy et al., 2007; Niyogi et al., 2010).

Several datasets of contemporary irrigated area have been developed at global and regional scales. The Global Irrigated Area Mapping project (GIAM; Thenkabail et al., 2009)

produced a global map of irrigated area with a 5 arcminute × 5 arcminute spatial resolution around the year 2000 based on various data sources including Advanced Very High Resolution Radiometer products and the Système pour l'Observation de la Terre Vegetation (SPOT VGT) product. Separate maps with a spatial resolution of 500m were produced for the Indo-Gangetic plains (Thenkabail et al., 2005), the Krishna basin in the Indian peninsula (Biggs et al., 2006) and the entire Indian subcontinent (Dheeravath et al., 2010) using Moderate Resolution Imaging Spectroradiometer (MODIS) data. Land use/land cover products derived from remote sensing have a high spatial resolution and the use of multitemporal satellite imagery enables the detection of land use/land cover change (Portmann et al., 2010). However, remote sensing approaches have a number of important limitations for spatio-temporal irrigated area mapping. Current classification algorithms cannot reliably distinguish between specific crop types or between irrigated and rainfed crops (Portmann et al., 2010). The fact that remotely sensed data only specifies the dominant land use/land cover class introduces a large amount of uncertainty to the resulting maps. For example, the total irrigated area in India according to GIAM is more than double the corresponding agricultural inventory data (Dheeravath et al., 2010). Classification algorithms for detecting irrigated crops, which must be trained on a large number of ground truth points, are also uncertain (Portmann et al., 2010). Moreover, the quality and availability of historical satellite data for India is poor (Houghton et al., 2012; Tian et al., 2014).

An alternative strategy to develop historical datasets of irrigated area is to use national and subnational agricultural census data. Freydank and Siebert (2008) combined data from the FAOSTAT database (FAO, 2015b) and other sources to create an annual time series of area equipped for irrigation at the country level for the period 1900–2003. This dataset has been used in several global studies to quantify the impact of irrigation on river discharge (Haddeland et al., 2007; Biemans et al., 2011; Pokhrel et al., 2012), surface fluxes and climate (Puma and Cook, 2010), water extractions (Gerten et al., 2008; Yoshikawa et al., 2014) and water storage (Döll et al., 2012). The majority of these studies used a scaling factor to disaggregate the country level data to a spatial grid, a requirement of Earth system models, resulting in considerable inaccuracies in large countries such as India. Recently, Siebert et al. (2015) developed a global dataset showing area equipped for irrigation between 1900–2005 with decadal time steps. This dataset was produced by downscaling national and subnational statistics on area equipped for irrigation using the HYDE database to define historical cropland extent and the Global Map of Irrigated Area (GMIA Siebert et al., 2005), a map showing the area equipped for irrigation around the year 2000, to define the spatial distribution of irrigated area. Area equipped for irrigation is useful for water resource managers and policy-makers because it gives an indication of the maximum capacity of the agricultural system and highlights areas for future development. The area actually irrigated is usually less than the area equipped for irrigation because of

land that is left fallow or devoted to rainfed agriculture in certain years, which may be influenced by the weather, or because of the failure of irrigation equipment or infrastructure (Siebert et al., 2005; Salmon et al., 2015). Historical information about the area actually irrigated is important because it allows water resource managers to understand how demand changes in response to variations in supply at various spatial and temporal scales (Siebert et al., 2015). In addition, hydrological and Earth system models that simulate irrigation processes require the area actually irrigated, rather than the area equipped for irrigation, as an input variable because irrigated area is usually represented as a separate land use (Nazemi and Wheater, 2015a). Lastly, data about the area actually irrigated is usually reported with the crop type, allowing a more detailed estimation of water demand taking into account specific crop water requirements.

This chapter presents a new spatio-temporal dataset to extend and improve existing information on the historical evolution of irrigated area in India. District-level agricultural inventory data showing the area actually irrigated of 25 crop types between 1956–2010 was spatially disaggregated using historical, spatially explicit maps of cropland extent and maps of crop suitability from the Food and Agriculture Organization Global Agroecological Zones database (IIASA, 2012). The resulting dataset has a 5 arcminute $\times$ 5 arcminute spatial resolution and provides a map for each irrigated crop type on an annual basis.

4.2 Methodology

4.2.1 Input data

Administrative boundaries

In common with the land use change modelling procedure outlined in Chapter 3, the present methodology requires the geographical area of each district unit to be consistent during the study period. Therefore, the map of district boundaries developed in Chapter 3 by modifying version 2.7 of the Global Administrative Areas Database (GADM, 2015) was also used here.

Agricultural inventory data

District-level agricultural inventory data specifies the annual irrigated area by crop type and irrigation source between 1956-2010. The data was obtained from the same sources as the land use/land cover inventory data described in Chapter 3, namely Indiastat (Indiastat, 2016) and, for 19 States between 1966–2009, the Icrisat Village Dynamics in South Asia (VDSA) project (Icrisat, 2016). Agricultural inventory data is collected in the same

way as land use/land cover data and therefore Figure 3.1, which groups India's States and Union Territories according to the way in which data is collected, also applies to agricultural inventory data.

Figure 4.1 shows the mean availability of Indiastat agricultural inventory data in each district unit for 5 year time slices. Overall, the data availability in district units surveyed using the first data collection method ($n = 251$, $\mu = 47.42$, $\sigma = 16.25$), where surveys are based on cadastral information, is higher than in districts surveyed using methods two ($n = 32$, $\mu = 18.67$, $\sigma = 8.01$) or three ($n = 13$, $\mu = 7.67$, $\sigma = 5.52$), although this varies at different times during the study period. The Indiastat inventory dataset does not include data points for the period between 1993–1997, as shown in Figure 4.1. However, this period is included in the Icrisat dataset. For areas with consistently poor data availability and quality, such as the western States of West Bengal and Assam, Agricultural Census data was used instead of district-level inventory data as input to the spatial disaggregation procedure. These censuses, which form part of the FAO World Programme for the Census of Agriculture (FAO, 2015c), provide state-level agricultural data at approximately five-year intervals between 1970–2010.

Cropland extent

Maps of cropland extent were obtained from the spatio-temporal land use/land cover change dataset developed in Chapter 3, which provides annual maps showing the fraction of each 5 arcminute $\times$ 5 arcminute grid cell devoted to cropland for the period 1956–2010.

Crop suitability

Crop suitability maps were obtained from the FAO Global Agro-ecological Zones (GAEZ) database, version 3.0 (IIASA, 2012). These maps, which have a 5 arcminute $\times$ 5 arcminute spatial resolution, quantify the extent to which prevailing soil, terrain and climatic conditions meet specific crop requirements under rainfed and irrigated conditions. The maps are computed for high, intermediate and low input levels, where input level refers to the intensity of agricultural production in terms of the prevalence of improved or high-yielding crop varieties, application of fertilizer and other chemicals and the extent of mechanization. They are further divided into maps considering baseline climatic conditions, taken as the average climate between 1961–2000, and three future time periods. To create the maps, crop suitability is computed at a 30 arcsecond $\times$ 30 arcsecond spatial resolution and aggregated to the 5 arcminute $\times$ 5 arcminute resolution by computing the mean suitability of cells at the fine resolution which were devoted to cropland according to the GLC2000

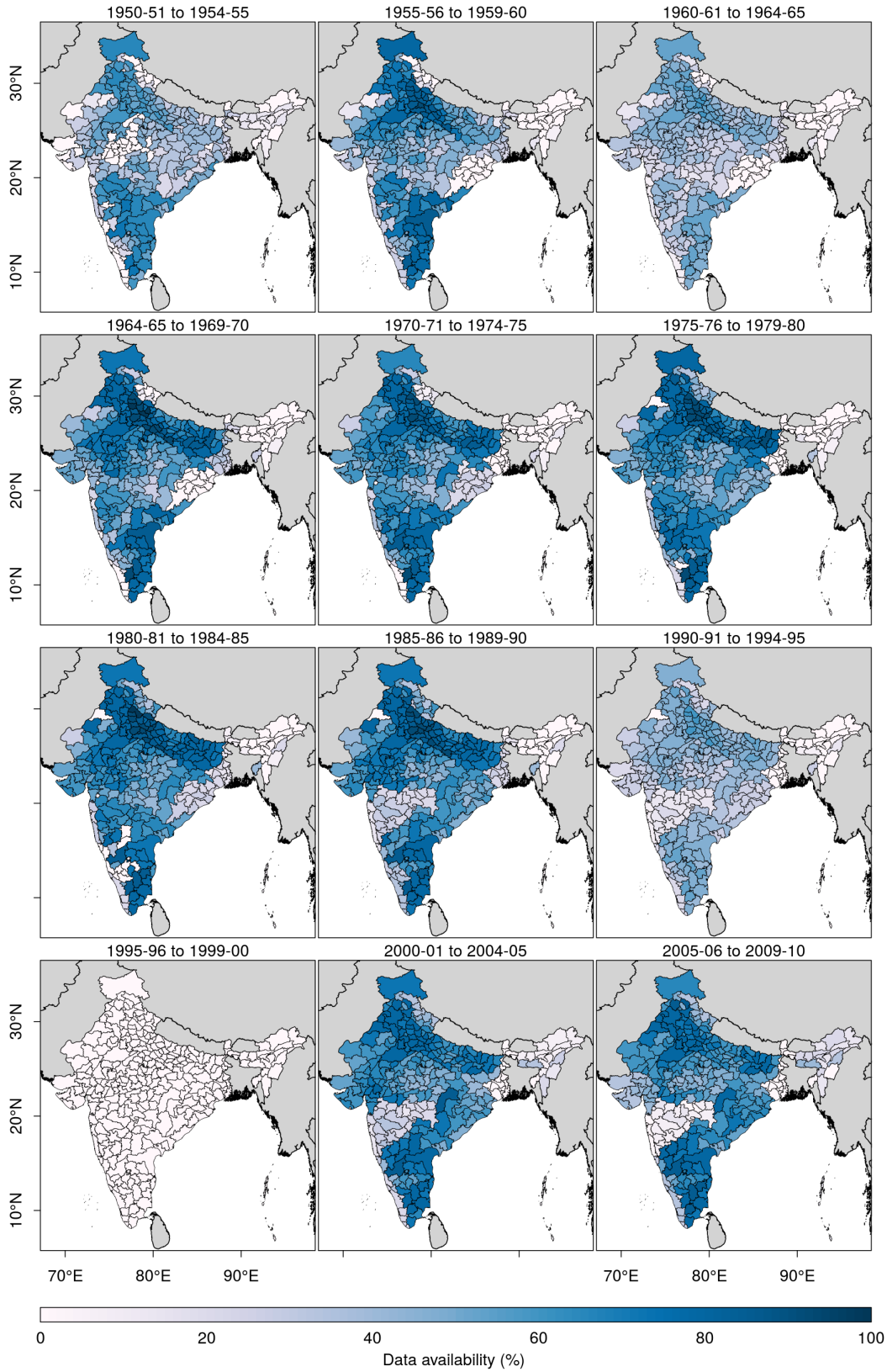


Figure 4.1.: Mean Indiastat data availability in each district unit over 5 year time slices. Data availability during the 1990s is poor because the Indiastat dataset does not provide data for the period 1993–1997. However, for most districts it was possible to infill this period using ICRISAT data.

land cover dataset (Bartholomé and Belward, 2005). The GAEZ methodology is based on the Harmonised World Soil Database (HWSD; FAO/IIASA/ISRIC/ISS-CAS/JRC, 2012), a dataset which includes several variables related to soil suitability for crop production. The maps used in the present study provide suitability values for irrigated crops under baseline climate conditions with high input level, which assumes agricultural production is primarily market oriented, utilisting high yield crop varieties, fertiliser and chemical pest, disease and weed control.

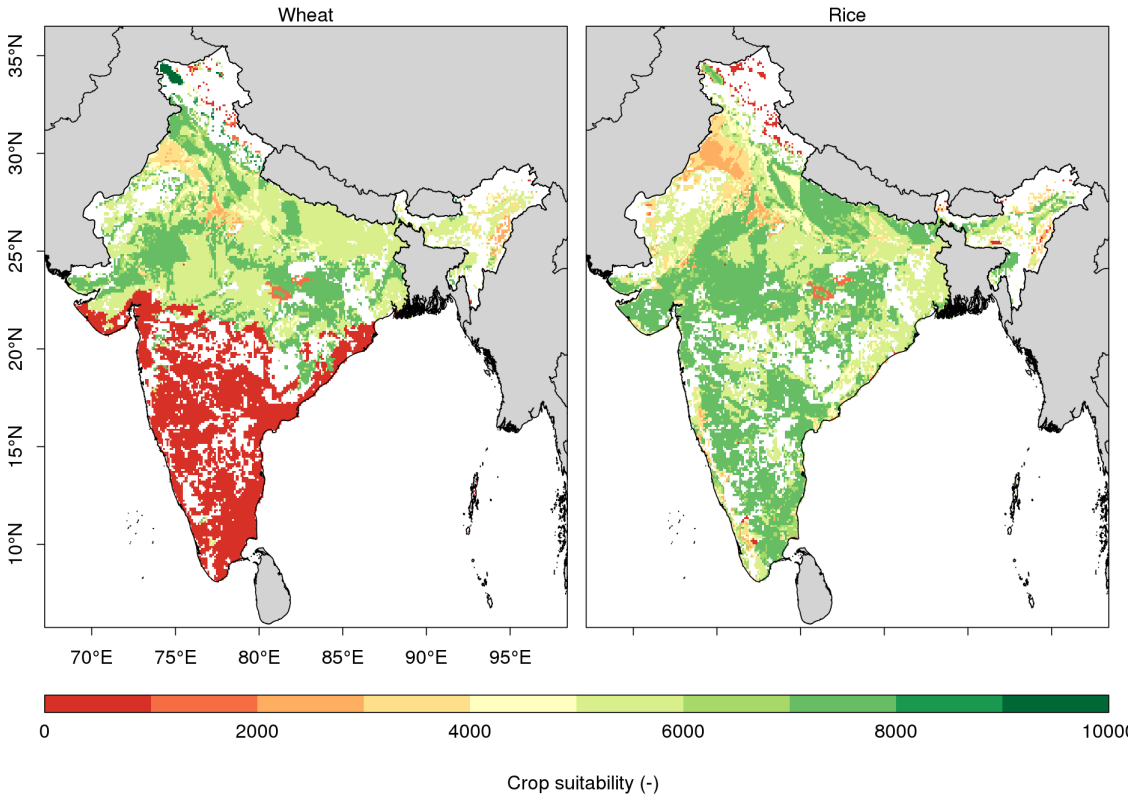


Figure 4.2.: Suitability maps for high-input irrigated wheat and rice from FAO GAEZ database.

4.2.2 Data preparation

In common with the land use/land cover inventory data encountered in Chapter 3, the district-level agricultural inventory contained various inconsistencies that were addressed using the same techniques described previously. State-level agricultural censuses, providing irrigated area data for several north-eastern states including West Bengal and Assam, were manually digitised and mapped to a format consistent with district-level inventory data. Maps of crop suitability were reprojected to Albers Equal Area projection using bilinear interpolation. The resulting maps had a spatial resolution of approximately 9km $\times$ 9km, consistent with the maps of cropland extent, ensuring that each grid cell in the study region had the same geographical area. In some cases there was no corresponding crop suitability map for crop types included in the agricultural inventory data. For this reason the suitability map for irrigated fruit was prepared by computing the mean suitability to banana, citrus, coconut and cacao, while the suitability map for vegetables was the mean suitability of cabbage, carrot, onion, tomato, white and sweet potato. In addition, the same crop suitability map was used for bajra (pearl millet) and ragi (finger millet) because the FAO GAEZ database does not separate different types of millet. For some minor crops it was not possible to assign a crop suitability map and in these cases uniform suitability was assumed. Table 4.1 shows the crop types mapped and the corresponding suitability map. The raster maps showing the fractional membership of each grid cell to the respective district units, developed in Chapter 3, were also used in the present analysis to define the grid cells to which district-level inventory data was assigned in the spatial disaggregation procedure.

4.2.3 Spatial disaggregation

The procedure to spatially disaggregate district-level inventory data is based on the assumption that the fraction of cropland devoted to each irrigated crop type is proportional to its suitability according to the FAO GAEZ crop suitability maps. Furthermore, whereas in other inventory-based gridded irrigated area datasets the area of irrigated crops in each grid cell is constrained by a map defining the area equipped for irrigation (e.g. Portmann et al., 2010; Siebert et al., 2015), in the present study it is assumed that irrigation water is available in all grid cells. This assumption is considered reasonable for India because of the widespread use of minor irrigation schemes, especially tubewells, which provide a cheap and readily available source of water in most parts of the country (Shah et al., 2006), particularly the most intensively irrigated regions in northern India. The spatial disaggregation algorithm deals with each district in turn, using the raster images of district units to define the cells to which district-level inventory data should be apportioned. The disaggregation procedure described below is visualised in Figure 4.3 for an example district,

Table 4.1.: Crop types included in the present study and the corresponding GAEZ suitability map.

Crop type	FAO GAEZ Suitability map
Wheat	Wheat
Maize	Maize
Rice (Autumn)	Rice
Rice (Winter)	Rice
Rice (Summer)	Rice
Barley	Barley
Bajra	Millet
Ragi	Millet
Sorghum (Kharif)	Sorghum
Sorghum (Rabi)	Sorghum
Other cereals (Kharif)	Rye, foxtail millet, oat, buckwheat
Other cereals (Rabi)	Rye, foxtail millet, oat, buckwheat
Gram	Gram
Tur	Tur
Other pulses (Kharif)	Phaseolus bean, chickpea, cowpea, dry pea
Other pulses (Rabi)	Phaseolus bean, chickpea, cowpea, dry pea
Sugarcane	Sugarcane
Condiments and spices	-
Fruit	Banana, citrus, coconut, cacao
Vegetables	Cabbage, carrot, onion, tomato, white potato, sweet potato
Other food crops	-
Groundnut	Groundnut
Sesamum	-
Rapeseed	Rapeseed
Linseed	-
Other oilseed	-
Cotton	Cotton
Tobacco	Tobacco
Fodder	-
Other non-food crops	-

Allahabad, located in the northern state of Uttar Pradesh.

For cell i, time j and crop k in a given district unit, overall suitability $suit_{i,j,k}$ is given by

$$suit_{i,j,k} = gaez_{i,k} \times crop_{i,j} \times dist_i \tag{4.1}$$

where *gaez* is crop suitability, *crop* is crop fraction and *dist* is district membership. The FAO GAEZ crop suitability maps do not include estimates of crop suitability outside the main growing regions of the respective crop types. For example, the suitability maps

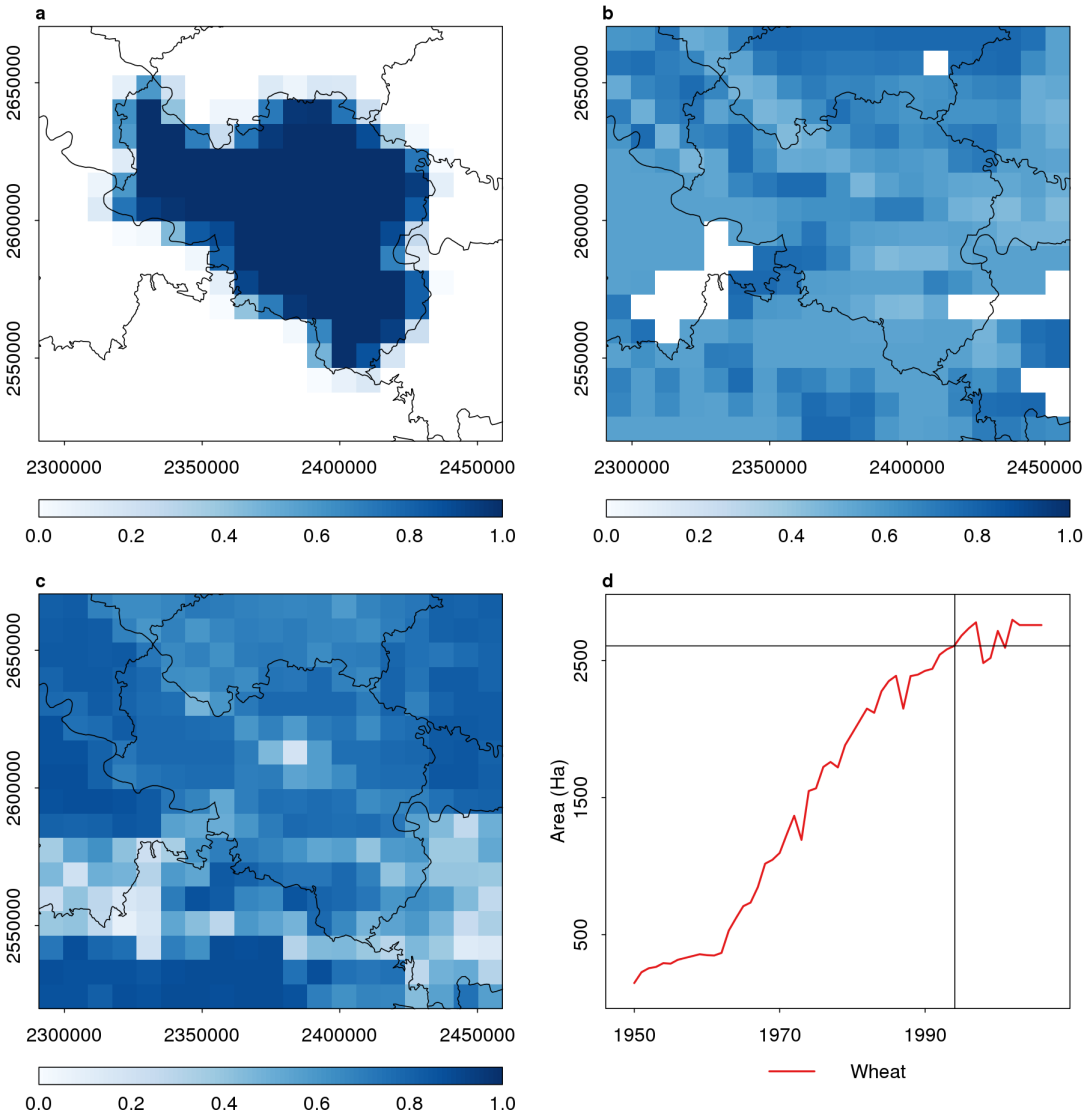


Figure 4.3.: Input data to disaggregation method for Allahabad district: **a** Fractional membership of each pixel to the district; **b** FAO GAEZ suitability of each pixel to irrigated, high-input wheat; **c** Fraction of each pixel devoted to cropland in the year 2000; **d** Change in area of irrigated wheat over the course of the study period.

of wheat and rice, shown in Figure 4.2, do not contain data for the Indian peninsula, where the production of these crops is small in comparison to the production in the Indo-Gangetic plains. Nevertheless crops are grown in locations other than their respective growing regions, albeit in relatively small quantities. In these cells the overall suitability, *suit*, is calculated using the crop fraction only. For a district with N grid cells, total suitability is transformed to a weighting factor, w, where

$$w_{i,j,k} = \frac{suit_{i,j,k}}{\sum_{i=1}^{N} suit_{i,j,k}}, \tag{4.2}$$

such that

$$\sum_{i=1}^{N} w_{i,j,k} = 1. \tag{4.3}$$

Irrigated area $A_{j,k}$, retrieved from the agricultural inventory data, is disaggregated by multiplying it by the weighting factor in each cell:

$$a_{i,j,k} = w_{i,j,k} \times A_{j,k} \tag{4.4}$$

and

$$\sum_{i=1}^{N} a_{ijk} = A_{jk} \tag{4.5}$$

where a_{ijk} is the area of crop k in cell i at time point j for the district under consideration. In some cases this results in pixels with an irrigated fraction greater than the crop fraction. Adopting the approach of Siebert et al. (2015), two versions of the dataset were produced. The first version prioritises consistency with the maps of crop fraction whereas the second prioritises consistency with the district level agricultural statistics, as follows

$$a_{i,j,k,crop} = \min(a_{i,j,k}, crop_{i,j}) \tag{4.6}$$

$$a_{i,j,k,stat} = a_{i,j,k} \tag{4.7}$$

where $a_{ijk,crop}$ is the adjusted crop area to maximise consistency with the corresponding crop fraction, $a_{ijk,stat}$ is the unadjusted crop area which maximises consistency with the agricultural statistics.

4.3 Results

Figure 4.4 shows the growth in irrigated area of six major crops at the national level, highlighting the dominance of irrigated wheat and rice. Figure 4.5 shows the growth in irrigated wheat has occurred mainly in the north and north-west of India, particularly in the states of Punjab, Haryana and western Uttar Pradesh. The rate of expansion of irrigated wheat was highest between 1965–1980, during which time the irrigated area increased threefold as the green revolution took hold. Figure 4.6 shows that while the area of irrigated rice has increased over the Indo-Gangetic plains it has also expanded in other parts of the country, particularly in eastern states such as Tamil Nadu, Andhra Pradesh, Orissa and West Bengal. Aside from the increase in area of irrigated wheat and rice, which was the main consequence of the green revolution, the expansion of other irrigated crops has been important in different parts of the country. For example, Figure 4.8 shows the growth in area of irrigated sugarcane, a perennial crop, during the course of the study period is centred around western districts of Uttar Pradesh. Comparing these maps with the maps of irrigated wheat and rice it can be seen that where irrigated sugarcane is grown the irrigated area of wheat and rice is less than in other parts of the Indo-Gangetic plains.

Validation of the irrigated area dataset is challenging because of the lack of independent datasets available for comparision. Irrigated area maps for the year 2000 were compared with the corresponding maps belonging to the MIRCA2000 dataset (Portmann et al., 2010), which were aggregated to the annual level using crop calendars provided with the dataset. The MIRCA2000 allocation procedure assumes that irrigation only occurs in areas equipped for irrigation defined by the GMIA dataset (Siebert et al., 2005), whereas in the present study it is assumed that all grid cells containing cropland have access to water for irrigation. Nevertheless, it can be seen that there is good overall agreement between the two datasets for the crop types shown.

4.4 Discussion

The gridded dataset was created by spatially disaggregating district-level inventory data containing irrigated areas of 25 major crops to a 5 arcminute $\times$ 5 arcminute spatial grid. Its intended application is in hydrological and Earth system modelling studies that require spatially explicit information about irrigated area to more accurately quantify the effects of irrigation on hydrological variables including groundwater level, streamflow and near-surface soil moisture. In addition, historical irrigated area maps are essential to identify feedback mechanisms between anthropogenic activities and the hydrological cycle (Nazemi and Wheater, 2015a). The spatial disaggregation methodology includes some important novel aspects compared to previous inventory-based efforts to map irrigated area (e.g.

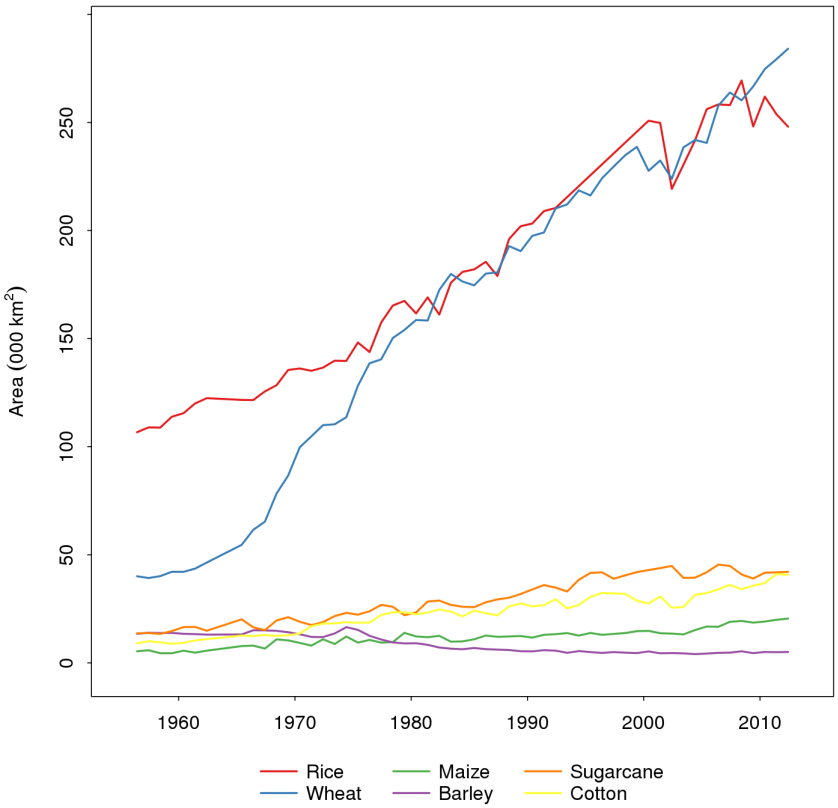


Figure 4.4.: Change in irrigated area of various crops in India at the national level between 1950-51 and 2010-11.

Siebert et al., 2005; Portmann et al., 2010; Siebert et al., 2015). Firstly, since it relies on district-level agricultural inventory data, the uncertainty in the spatial disaggregation procedure is substantially reduced compared to similar approaches using state or national level inventory data (e.g. Siebert et al., 2015). This is particularly relevant in the large northern state of Uttar Pradesh, which contains a large proportion of the fertile Indo-Gangetic plain, where there is substantial spatial variation of irrigation intensity between eastern and western districts. A further advantage of the present dataset is the inclusion of FAO GAEZ crop suitability maps in the spatial disaggregation procedure, which describe the biophysical suitability of grid cells to individual crops and provide a physical basis for the resulting spatial pattern of irrigated area. This addresses one of the main weaknesses of previous inventory-based irrigated area datasets, which do not consider biophysical constraints during the spatial allocation of irrigated area (e.g. Portmann et al., 2010). The fact that the dataset shows the area actually irrigated, rather than area equipped for irrigation, means that it is more suitable for hydrological modelling applications than several previous datasets because it is directly related to the volume of water applied (Siebert et al., 2015).

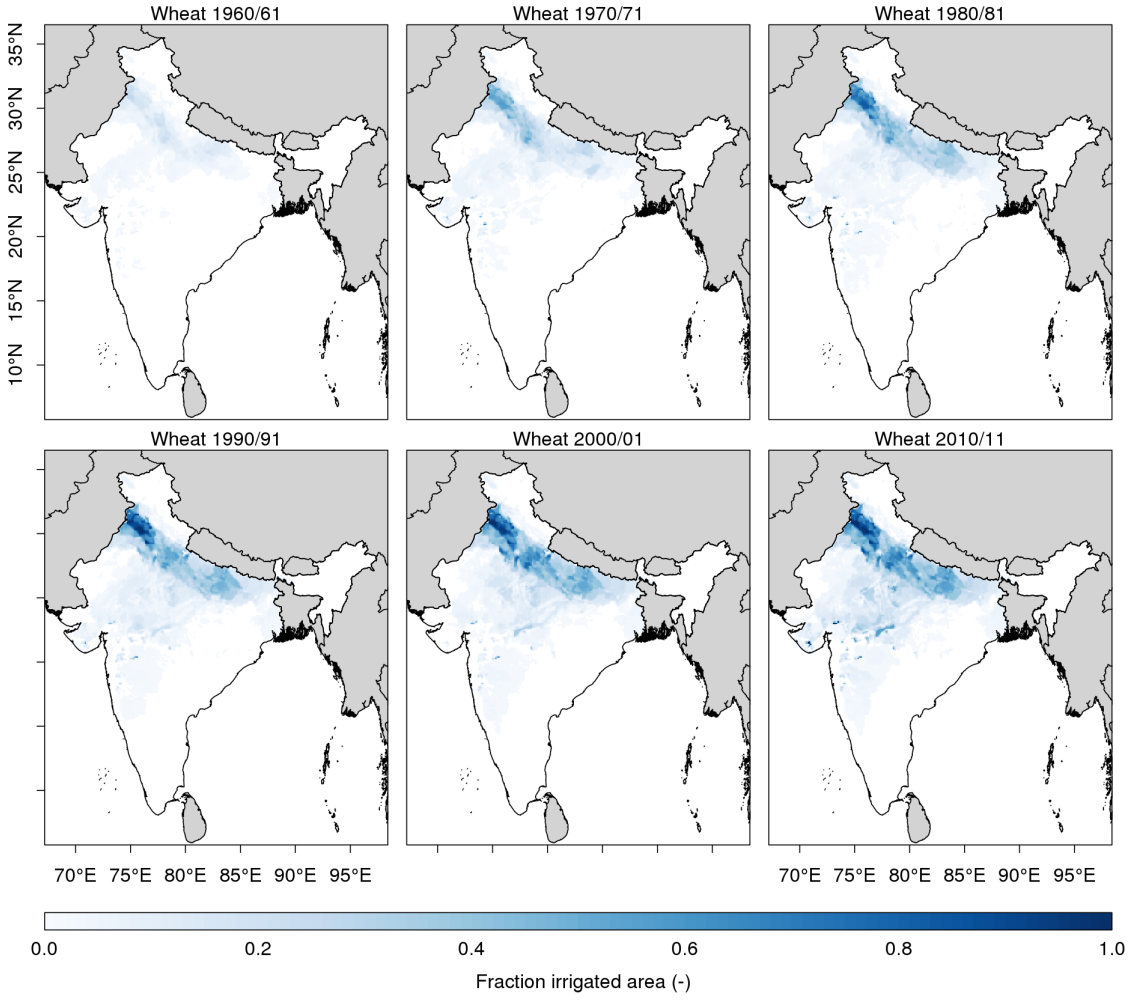


Figure 4.5.: Irrigated wheat area at various time points during the study period.

Furthermore, providing the irrigated area of individual crops allows parameterisations that take into account specific crop characteristics.

The gridded irrigated area dataset is associated with a large amount of uncertainty from various sources. Uncertainty in the inventory data is related to the survey methods employed to collect the data. Figure 4.1 shows the majority of States and Union Territories, including the States in northern India where the green revolution primarily took place, employ full annual surveys of agricultural variables which is the most accurate data collection of the three methods employed across India. However, despite considerable efforts to remove noise from the district-level inventory data it is likely that some remains. As

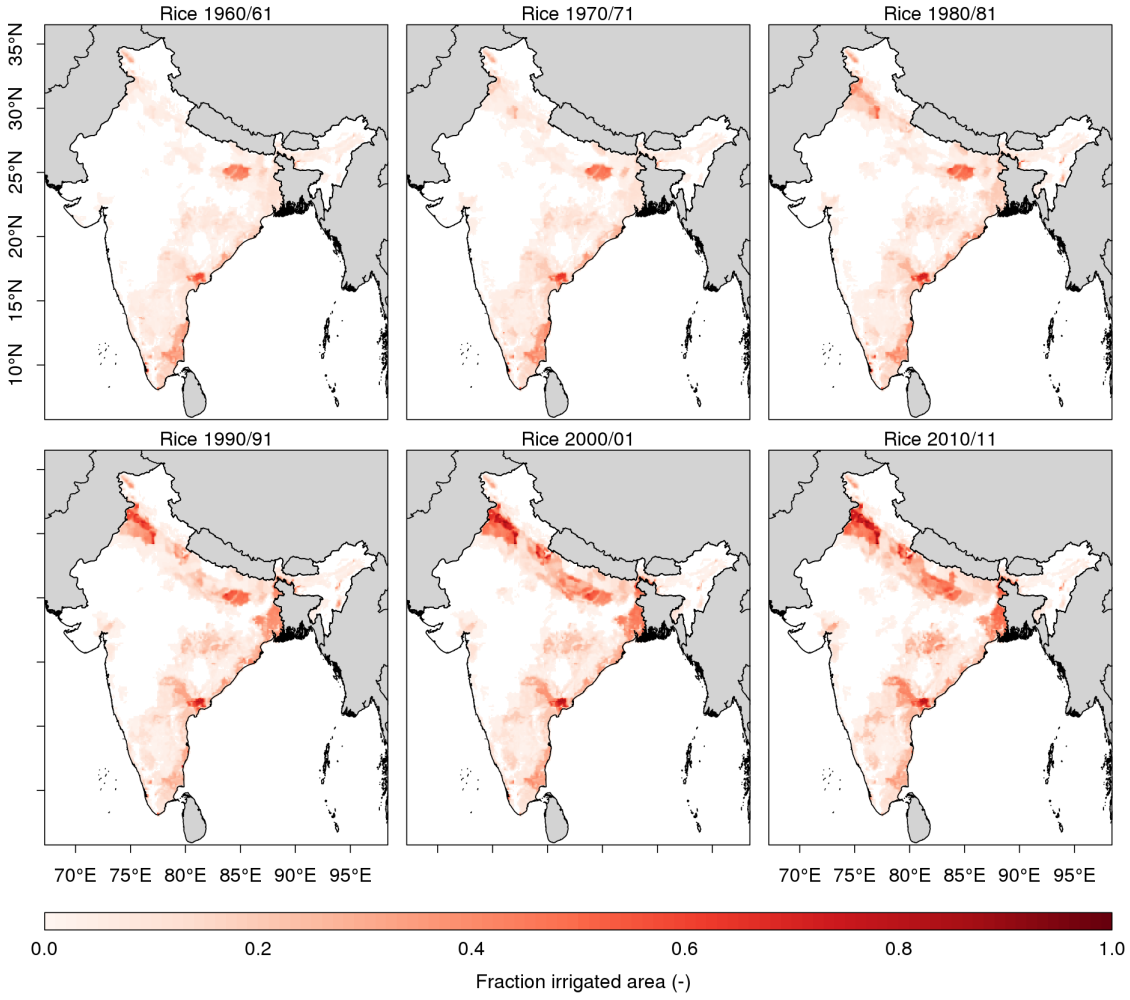


Figure 4.6.: Irrigated rice area at various time points during the study period.

discussed in Chapter 3, there is also uncertainty about the administrative boundaries derived from the GADM dataset. At the level of individual grid cells uncertainty is related to the spatial input data as well as the spatial disaggregation algorithm. Maps of crop fraction maps inherit uncertainty from the GlobeCover30 land cover product, uncertainty from the land use/land cover inventory data as well as structural uncertainty from the CLUE model itself. Maps of crop suitability from FAO GAEZ are subject to uncertainty from the GLC2000 land cover dataset because this dataset determines the cells at the fine resolution that are considered when calculating the mean suitability at the coarse resolution. Furthermore, the Harmonised World Soil Database, upon which the FAO GAEZ suitability maps are based, is highly uncertain over India because of the lack of local soil

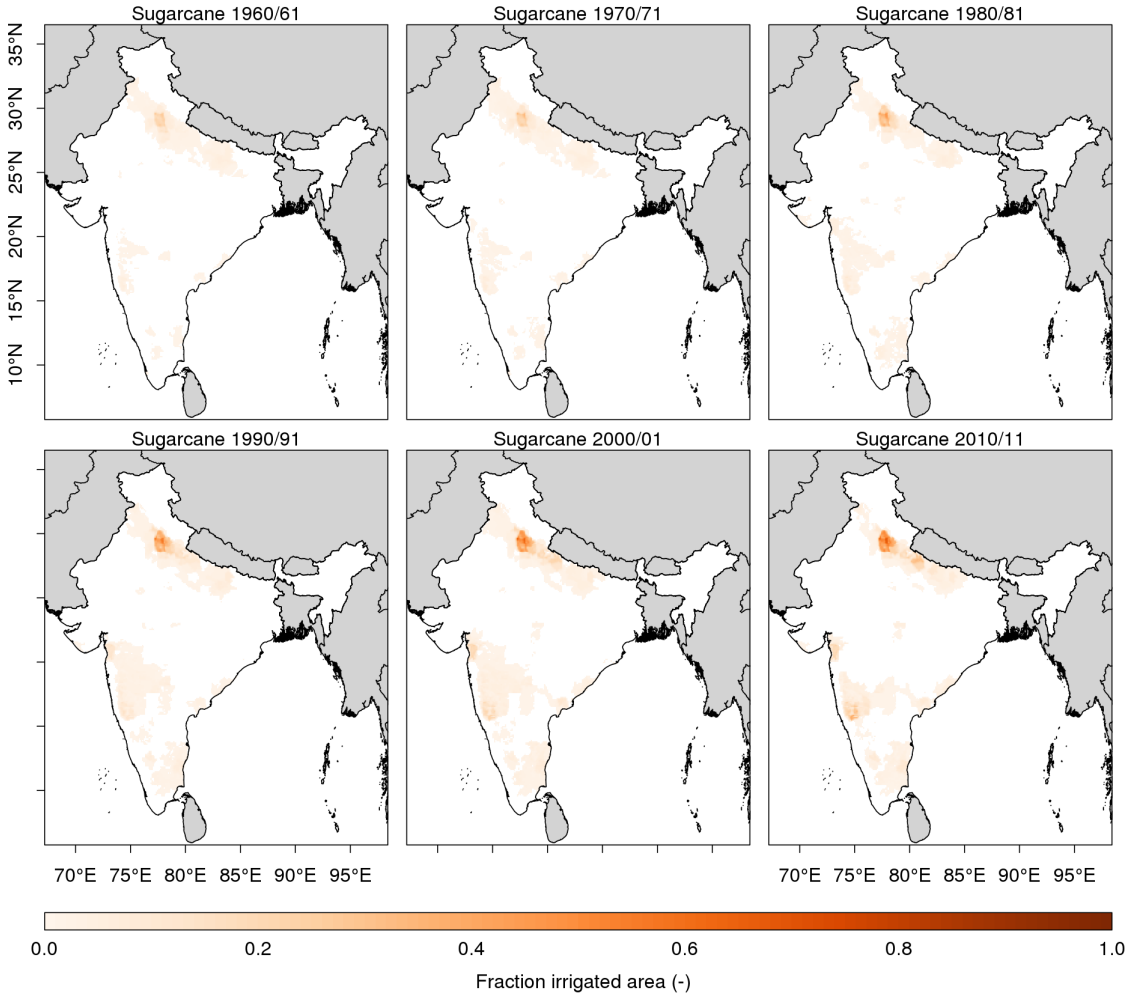


Figure 4.7.: Irrigated sugarcane area at various time points during the study period.

information. Moreover, the suitability maps have not been calibrated to observations of irrigated area because no such observational dataset exists. Remote sensing imagery provides the only practicable means of deriving this sort of data at a regional scale and there have been some attempts to classify satellite images according to crop type for parts of India (Singh et al., 2011), although the data is not currently in the public domain. As remote sensing technology and processing algorithms improve, data showing the spatial distribution of crop types will become more widespread. These datasets could be used to derive alternative crop suitability maps based on statistical analysis to quantify the relationship between the spatial distribution of irrigated crops and biophysical suitability from the FAO GAEZ dataset as well as socioeconomic variables such as population density

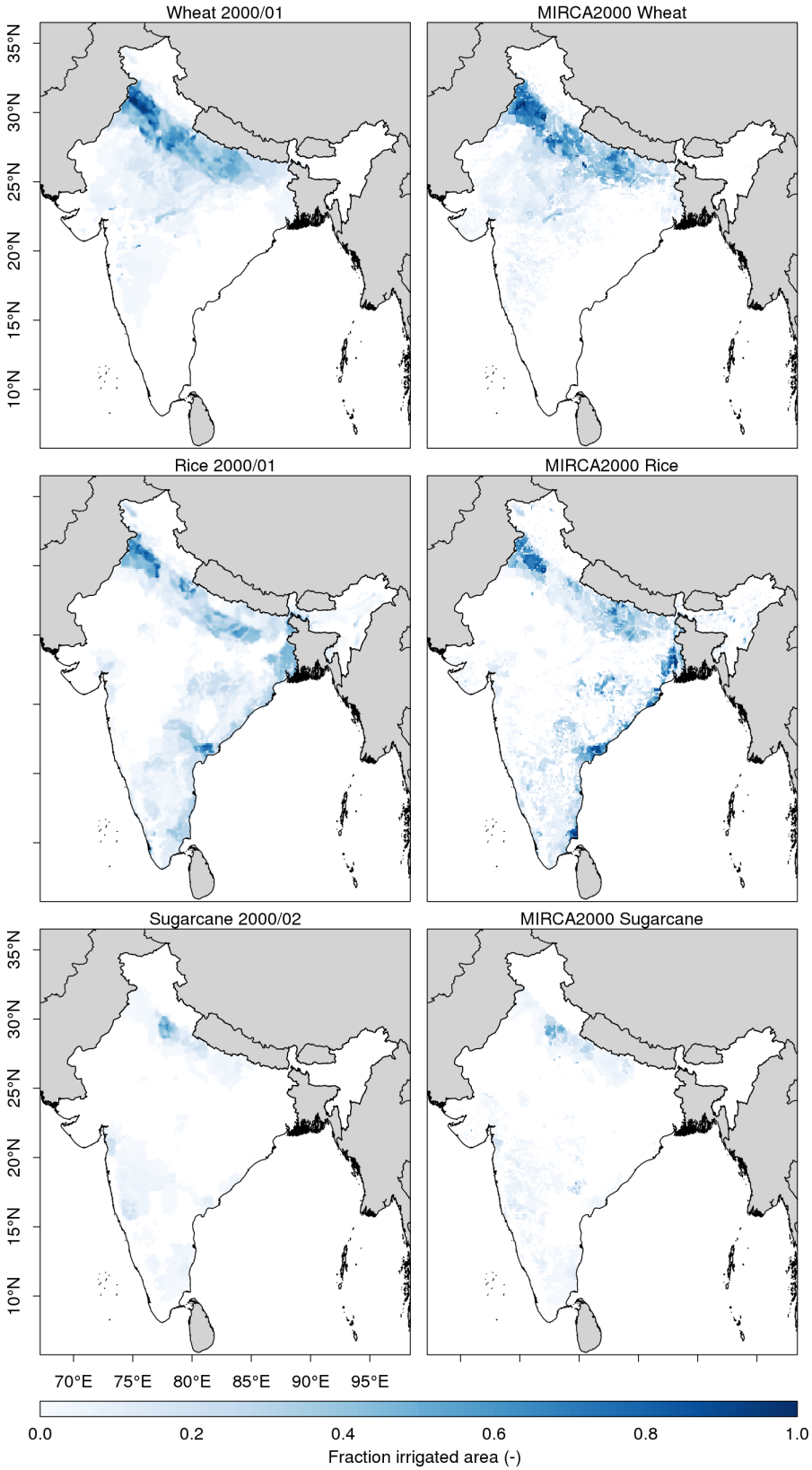


Figure 4.8.: Comparison between maps of irrigated wheat, rice and sugarcane from the current study and corresponding maps from the MIRCA2000 dataset.

and access to infrastructure. While these maps could be supplied to the disaggregation procedure described above, they could also be used as an input to dynamic land use change models to simulate competition between irrigated crop types.

The spatial disaggregation procedure employed in the present study recognises that more irrigation will take place in areas of high biophysical suitability than in areas of low biophysical suitability. However, the assumption that the spatial distribution of irrigated area is directly proportional to the relative suitability value is clearly a simplification. Furthermore, while the assumption that all grid cells have equal access to water for irrigation is considered reasonable for many parts of India, it is highly probably that some grid cells to which irrigated area has been assigned do not, in reality, have access to water for irrigation. Furthermore, the methodology does not account for social and cultural factors that may influence farmer decisions about the quantity and location of irrigated crops (O'Keeffe et al., 2015). Commenting on the MIRCA2000 dataset, Portmann et al. (2010) observed, "human decisions on crop production are based on complex reasoning that cannot be captured by macroscale modelling approaches". For these reasons it is not advised to use the dataset for local-scale studies in which the analysis is sensitive to spatial distribution of irrigated areas. Rather, the dataset is suited to regional-scale investigations which require a general sense of the spatial distribution of irrigated crops but which do not depend on the accuracy of irrigated area shown in individual pixels.

The irrigated area dataset does not temporally disaggregate annual crop data to specific growing seasons. This is because different applications of the dataset will have different requirements in terms of the sub-annual temporal resolution. Hence, it is possible for the total irrigated area in each grid cell to be greater than the crop fraction. This frequently occurs in northern India where multiple cropping practices, particularly rice-wheat systems, are widespread (Thenkabail et al., 2005). Additional data processing could apply a temporal disaggregation procedure to generate maps at a higher temporal resolution. For example the MIRCA2000 dataset used crop calendars from AQUASTAT (FAO, 2015a) and expert knowledge to disaggregate the irrigated area of 26 crops to a monthly temporal resolution (Portmann et al., 2010). Alternatively a dynamic vegetation model such as LPJmL (Bondeau et al., 2007; Rost et al., 2008) could be used to simulate crop growth cycles based on spatially explicit input data climatic conditions, soil properties and agricultural management practices (Portmann et al., 2010). Similarly, the gridded dataset does not separate irrigated area by source. However, while the harmonised database of agricultural statistics provides information about the fraction of total irrigated area from different sources, the type of disaggregation procedure employed will depend on the objectives of individual studies and, consequently, it was not considered necessary or appropriate to undertake this sort of analysis in the present study.

4.5 Chapter summary

This chapter described a new spatio-temporal dataset showing the irrigated area of 25 crop types in India between 1956–2010. The dataset was developed by spatially disaggregating district-level agricultural inventory data to a 5 arcminute × 5 arcminute spatial grid using maps of historical cropland extent, developed in Chapter 3, and biophysical crop suitability obtained from the FAO GAEZ database. One of the main strengths of the methodology compared to previous attempts to map historical irrigated area is the fact that it considers the biophysical suitability of grid cells to individual crops, providing a physical basis for the spatial allocation of district-level agricultural inventory data. Moreover, in contrast to several previous irrigated area datasets which show the area equipped for irrigation, the dataset developed here shows the area actually irrigated; a quantity which is more closely related to water consumption. Lastly, the dataset provides the irrigated area of 25 crop types, allowing modellers to take into account the physical characteristics of individual crops. The irrigated area dataset contains considerable uncertainty arising from the input datasets as well as the spatial disaggregation procedure, which is based on the simple assumption that the fraction of cropland irrigated is proportional to its suitability to various irrigated crops. To improve the methodology the crop suitability maps, which are currently derived from theoretical considerations only, should be calibrated to local conditions. This will depend on the future availability of remote sensing products which are capable of reliably identifying specific irrigated crops.

The intended use of the irrigated area dataset is for hydrological and Earth system modelling applications that require spatially explicit information about irrigated area to quantify the effects of irrigation on regional hydrology. Chapter 5 incorporates the land use/land cover change dataset developed in Chapter 3 with a state of the art land surface model in order to investigate the impact of land use/land cover change across northern India on the behaviour of the South Asian monsoon. Since the current generation of land surface models do not include irrigation processes, simulated soil moisture values are post-processed using the irrigated area dataset in order to additionally investigate the impact of irrigation on the behaviour of the monsoon.

5 Quantifying the impact of land change on the South Asian monsoon

Large-scale environmental change across the Indo-Gangetic plains as a result of India's green revolution may influence the behaviour of the South Asian monsoon in complex ways. To date, however, the lack of appropriate spatio-temporal data about land change has ensured that assessments of the effects of environmental change on the monsoon circulation are associated with considerable uncertainty. This chapter starts by describing the various land-atmosphere coupling mechanisms and the way in which these are influenced by changes in land use/land cover and irrigated area. Based on a critical review of the various methodologies for incorporating irrigation processes in climate modelling experiments, a new methodology is devised which exploits the novel characteristics of the datasets developed in previous chapters. Firstly, the land use/land cover change dataset developed in Chapter 3 is used to constrain the Joint UK Land Environment Simulator (JULES; Best et al., 2011; Clark et al., 2011a), a process-based land surface model, to assess the impact of land use/land cover change on water resources in northern India. Simulated values of soil moisture are then post-processed to account for irrigation activities using the irrigated area dataset described in Chapter 4. This results in two historical soil moisture datasets: one considering land use/land cover change only, and one additionally considering the growth in irrigated area. These datasets were subsequently used by other members of the Hydroflux India consortium to force two regional climate models in an experiment to assess the impact of land change on the South Asian monsoon.

5.1 Literature review

This section provides an overview of the basic mechanisms of land-atmosphere coupling and describes the state of the art in terms of understanding of the effects of irrigation on the behaviour of the South Asian monsoon. In addition, it critically reviews the various technical approaches for including the effects of irrigation in the lower boundary condition of global and regional climate models.

5.1.1 Surface energy and water balance

The global climate system is fundamentally driven by solar radiation, which almost all exists in the shortwave range (Barry and Chorley, 2010). Incoming radiation may be absorbed, reflected or transmitted by the atmosphere (Pitman, 2003; Seneviratne et al., 2010; Barry and Chorley, 2010). The net radiation at the Earth's surface, R_n, is expressed as follows:

$$R_n = S\downarrow(1-\alpha) + L\downarrow - L\uparrow \tag{5.1}$$

where $S\downarrow$ is incoming shortwave radiation, α is surface albedo, and $L\downarrow$ and $L\uparrow$ are incoming and outgoing longwave radiation, respectively. Net radiation is transferred to the atmosphere primarily through sensible and latent heat fluxes (Pitman, 2003). Sensible heat is the energy transferred from the Earth's surface to the atmosphere by convection and conduction, while latent heat is the energy absorbed or released by the atmosphere when water changes state. The latent heat of vapourisation, which varies with temperature, is the amount of energy released by the atmosphere to evaporate a quantity of water. Correspondingly, when water condenses an amount of latent heat corresponding to the latent heat of vapourisation is transferred to the atmosphere (Barry and Chorley, 2010). The Bowen ratio, B, is the partition between sensible and latent heat fluxes:

$$B = \frac{\lambda E}{H} \tag{5.2}$$

where λE is latent heat flux and H is sensible heat flux. In addition to sensible and latent heat fluxes, net radiation drives the flux of heat to the soil and, where biomass is present, chemical energy which is stored in plants during photosynthesis and subsequently released during respiration (Barry and Chorley, 2010). Collectively, these terms gives rise to the surface energy budget equation for a surface soil layer:

$$\frac{dQ}{dt} = R_n - H - \lambda E - G - F \tag{5.3}$$

where dQ/dt is the total energy flux within the soil layer, G is soil heat flux and F is chemical energy storage. As the soil thickness decreases dQ/dt approaches zero and G

represents the ground heat flux at the land surface (Seneviratne et al., 2010).

Precipitation reaching the Earth's surface may be intercepted by vegetation or it may fall directly to the soil surface. Intercepted water either evaporates directly from the canopy or falls to the soil surface, where it may infiltrate the soil matrix or contribute to surface runoff. Infiltration may be evaporated from the surface, drain through the soil to recharge the underlying aquifer or drawn up by plant roots and transpired from the canopy (Pitman, 2003). Thus, for the same soil layer considered in Eq. 5.3, the surface water budget equation is written:

$$\frac{dS}{dt} = P - I - E - Q_s - Q_d \tag{5.4}$$

where dS/dt is the change in storage within the soil layer, P is precipitation, I is canopy storage or interception, E is evapotranspiration, Q_s is surface runoff and Q_d is subsurface drainage.

5.1.2 Role of vegetation in soil moisture feedbacks

The biosphere is the main interface between soil moisture and the atmosphere (Lawrence et al., 2007; Dirmeyer et al., 2006). Consequently, land cover and its temporal dynamics have important effects on the interaction between the two domains (Pielke et al., 2002; Feddema, 2005; Catalano et al., 2016). These may be caused by biogeochemical processes, which alter the chemical composition of the atmosphere by affecting the land surface carbon flux, or biogeophysical processes, which alter the surface energy and water balances by changing the surface albedo and Bowen ratio (Feddema, 2005). Surface albedo determines the total energy available at land surface (Meehl, 1994; Pitman, 2003). Forests tend to have an albedo between 0.09 and 0.18, while for cereal crops the value typically lies between 0.18 and 0.25 (Barry and Chorley, 2010). Plant physiology affects the Bowen ratio by influencing the rate of transpiration (Hillel, 1998). Canopy structure provides a further control on Bowen ratio through its effect on interception and bare soil evaporation, as well as influencing surface roughness, which affects the transfer of momentum from the surface to the atmosphere (Bounoua et al., 2002; Arneth et al., 2012). Irrigation changes the surface water balance and, therefore, alters the Bowen ratio (Boucher et al., 2004). It may also affect the surface albedo both directly, by increasing the amount of water land surface, and indirectly, by altering the physical characterisitics and seasonal dynamics of crops (Seneviratne et al., 2010).

5.1.3 Soil moisture-climate interactions and feedbacks

Soil moisture is defined as the water stored in the unsaturated zone of a soil layer (Hillel, 1998). Volumetric soil moisture, θ, defines the volume of water, V_w, compared to the total volume, V_t, of the soil layer, as follows:

$$\theta = \frac{V_w}{V_t} \tag{5.5}$$

where V_t is comprised of the volume of solids, V_s, volume of air, V_a and volume of water. Soil porosity, ϕ, is the theoretical maximum value of volumetric soil moisture (Shaw, 1994). Field capacity, θ_{fc}, is defined as the maximum amount of water that can be held by the soil matrix against the force of gravity, while the permanent wilting point, θ_{wilt}, is defined as the point at which plant roots cannot draw any water from the soil matrix (Shaw, 1994; Seneviratne et al., 2010).

Equations 5.3 and 5.4 show that the surface energy and water balances are linked through the evapotranspiration term. The evaporative fraction, EF, is a useful concept for describing the relationship between soil moisture and evapotranspiration:

$$EF = \frac{\lambda E}{R_n}. \tag{5.6}$$

As Figure 5.1 shows, the evaporative fraction increases with soil moisture until it reaches its maximum value, EF_{max}, corresponding to a critical soil moisture value, θ_{crit}, which lies between the permanent wilting point, θ_{wilt}, and field capacity, θ_{fc} (Seneviratne et al., 2010). For soil moisture values above θ_{crit} evapotranspiration is energy limited while for values less than θ_{crit} evapotranspiration is soil moisture limited. When soil moisture falls below θ_{wilt} evapotranspiration cannot take place. This gives rise to a transitional zone, $\theta_{wilt} \leq \theta \leq \theta_{crit}$, where soil moisture directly constrains evapotranspiration and consequently provides feedbacks to the atmosphere through its impact on the partition between sensible and latent heat (Koster et al., 2004, 2009; Lobell et al., 2009; Seneviratne et al., 2010).

5.1.4 Hydrometeorological feedbacks in the Indo-Gangetic plains

Several studies of the Global Land Atmosphere Coupling Experiment (GLACE) (Koster et al., 2004, 2006; Guo et al., 2006), based on an ensemble of twelve atmospheric general circulation models, identified the monsoon region in northern India, as well as the Sahel region in Africa and central North America, as a global "hot spot" of soil moisture-precipitation coupling strength during the boreal summer. Results from GLACE show that models with a strong link between soil moisture and the surface energy balance, and

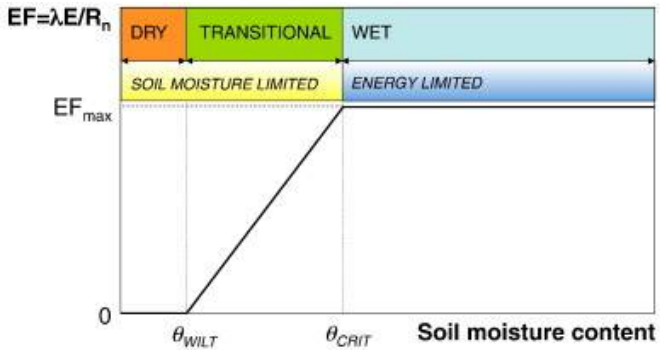


Figure 5.1.: Conceptual diagram showing soil moisture and energy limited evapotranspiration regimes (Seneviratne et al., 2010)

.

between the surface energy balance and precipitation, had more explanatory power than models with weak representations of either of these two feedback mechanisms. The hot spots identified by the GLACE experiments are characterised by hydroclimatic regimes in which soil moisture constrains evapotranspiration (Koster et al., 2009; Catalano et al., 2016). Increasing soil moisture through irrigation in transitional regions shifts the Bowen ratio from sensible to latent heating (e.g. Puma and Cook, 2010; Seneviratne et al., 2010). This leads to greater levels of evapotranspiration which, in turn, can enhance local convection and increase precipitation (Sacks et al., 2009; Puma and Cook, 2010). In India, Sen Roy et al. (2011) highlighted a trend of increasing dry-season, non-monsoonal precipitation across the Indo-Gangetic plains following the green revolution as a result of increasing evapotranspiration. However, Tuinenburg et al. (2014) showed that the direct influences of soil moisture on precipitation in India are highly uncertain and, in any case, of low overall importance compared to indirect effects arising from changes to the large-scale temperature contrast which is critical to the formation of the South Asian monsoon circulation (e.g. Singh, 2016). In monsoon regions the alteration of the temperature gradient between land and ocean as a consequence of irrigation has been shown to influence monsoon circulation (Pielke et al., 2011). The most intensive irrigation in India occurs across the Indo-Gangetic plains, where heating of the land surface during boreal spring is critical to establish the low-pressure trough which drives the initial monsoon circulation (Niyogi et al., 2010; Shukla et al., 2014).

Several studies have identified changes in the surface energy budget as a result of irrigation. For example, Sen Roy et al. (2007) showed a cooling of 0.34°C in maximum temperatures during the growing season in northern India between 1947–1964 (pre-green revolution) and 1980–2003, although the overall trend was not significant at the 95% confidence level. These results are confirmed by several global and regional modelling studies investigating the impact of irrigation on the surface air temperature (e.g. Lobell et al.,

2008, 2009; Sacks et al., 2009). Similarly, Douglas et al. (2006) used a water balance model to demonstrate that mean annual moisture fluxes across contemporary India had increased by 17% in total compared to pre-agricultural land cover, with a 7% increase in the monsoon season and a 55% increase in the dry season, when most irrigation takes place. According to Douglas et al. (2006), up to two-thirds of this increase could be attributed to irrigation with the remaining increase due to land cover change. The cooling of the land surface as a result of irrigation could mask surface warming that would otherwise be attributed to anthropogenic climate change (Cook et al., 2011). Pioneering work by Shukla and Mintz (1982), based on sensitivity experiments with a GCM, showed that dry conditions across India and South Asia increased the land-ocean temperature contrast, enhancing water vapour transport from the ocean and resulting in higher precipitation during the monsoon. Later, Douville et al. (2001) used a regional climate model to show that increasing soil moisture during March–May caused a shift in monsoon rainfall from eastern India to northern India. A sensitivity study by Asharaf et al. (2012), using the COSMO-CLM Earth system model, also showed that artificially increasing pre-monsoon soil moisture weakened the monsoon circulation compared to the control experiment. Recently, Unnikrishnan et al. (2015) demonstrated that the land-atmosphere coupling has an important influence on seasonal and intraseasonal variability of the South Asian monsoon. This study demonstrated that the maximum coupling strength between soil moisture and the monsoon circulation occurred over north-west and central India, supporting the findings of Koster et al. (2004).

Many researchers have extended these sensitivity experiments by attempting to isolate the effects of irrigation on the South Asian monsoon. Lee et al. (2009) found statistically significant correlation (Lee et al., 2009, Figure 3) between observed decreases in July rainfall across central and southern India between 1982–2003 and increases in the observed Normalized Difference Vegetation Index (NDVI) anomolies during March–May over the same period. Years with more vegetation cover during the pre-monsoon season were associated with weaker July rainfall, while areas with positive pre-monsoon NDVI anomolies were positively correlated ($r = 0.87$) with irrigated area from the Global Map of Irrigated Areas (GMIA) product (Siebert et al., 2005). Niyogi et al. (2010) assessed the causal relationship between irrigation during the pre-monsoon season and changes to the behaviour of the Indian monsoon, discovering a negative relationship between increased irrigation during March–April and July rainfall. Meanwhile, Dirmeyer et al. (2009) showed that precipitation is most sensitive to soil moisture during March–May and September–November by analysing several observational and reanalysis datasets. Several modelling studies have investigated the impact of irrigation on the South Asian monsoon. Saeed et al. (2009) demonstrated that including irrigation processes in the land surface scheme of the REMO Earth system model substantially reduced its warm bias over the Indian subcontinent,

resulting in a more realistic representation of the South Asian monsoon. More recently, Tuinenburg et al. (2014) used one global climate model and three regional climate models to show that extensive irrigation across the Indo-Gangetic plains resulted in changes to the large-scale monsoon circulation manifested as reduced precipitation in eastern India and increased precipitation in northern and western India. These results, which are consistent across all models in the study, confirm the results of previous sensitivity studies (e.g. Douville et al., 2001; Asharaf et al., 2012). Meanwhile, Shukla et al. (2014) showed that forcing a GCM with irrigation-adjusted soil moisture, as well as greenhouse gas emissions, improved the ability of the model to simulate observed trends in twentieth century monsoon rainfall compared to simulations which only included greenhouse gas emissions.

The current generation of Earth system models do not include irrigation processes (Nazemi and Wheater, 2015a). Instead, various approaches have been adopted to include irrigation in modelling workflows; some of which are summarised in Table 5.1. Several studies utilise the GMIA product (Siebert et al., 2005), which shows the area equipped for irrigation around the year 2000, to specify irrigated areas. Haddeland et al. (2007) scaled the GMIA map with national time series data about irrigated area from FAOSTAT (FAO, 2015b) in order to generate a spatio-temporal dataset of irrigated area for the twentieth century. In most of the studies that utilise the GMIA dataset it is assumed that the area equipped for irrigation is entirely utilised. This assumption most likely results in an overestimation of the total irrigated area because of agricultural land that is rainfed or left fallow in certain years (Döll et al., 2014; Siebert et al., 2015). Having defined irrigated area, there is little agreement about the most accurate way to simulate the effect of irrigation on surface soil moisture. The basic approach followed in the majority of cases is to artificially increase simulated soil moisture values in irrigated cells. Most studies have taken some fraction of the saturated water content of the soil profile. The most extreme approach was that of Lobell et al. (2006) who, assuming all cropland was irrigated, set irrigated regions to the saturated water content for the duration of the study period. Lobell et al. (2009) instead used an approach where the simulated soil moisture value was only adjusted if it fell below a certain critical value, allowing it to evolve freely otherwise. Boucher et al. (2004) and Douglas et al. (2006) instead simulated the moisture flux, setting evapotranspiration from irrigated areas to the potential rate. Puma and Cook (2010) and Shukla et al. (2014) used a spatio-temporal dataset of irrigated area for the period 1901–2000 developed by Wisser et al. (2010), in which the GMIA map is scaled with national time series data of irrigated area compiled by Freydank and Siebert (2008). This dataset was then supplied to WBMplus, a water balance and transport model which extends the original WBM model (Vörösmarty et al., 1998), to generate monthly soil moisture values based on various assumptions about crop water requirements.

The majority of studies included in Table 5.1 do not account for variability in water demand for irrigation between seasons (e.g. Lobell et al., 2009; Tuinenburg et al., 2014). Instead, many studies make the assumption that irrigation can occur throughout the year. For example, Lobell et al. (2009) increased soil moisture in irrigated areas whenever the simulated values fell below 40% of the saturated water content, irrespective of the time of year. One of the main limitations of this approach in India is that it would cause there to be more irrigation in the pre-monsoon season, when temperatures are at their highest, than would be likely to occur in reality because farmers do not usually grow crops during this season (Thenkabail et al., 2005; Douglas et al., 2006). As discussed previously, several studies have highlighted the sensitivity of the monsoon to soil moisture in the months leading up to the monsoon season (e.g. Dirmeyer et al., 2009; Lee et al., 2009; Niyogi et al., 2010). Thus, it is critically important that methods to incorporate the effects of irrigation in the lower boundary condition of atmospheric models account for the spatial and temporal variability of growing seasons of specific crops. A simple strategy adopted by Douglas et al. (2006) was to adjust the irrigated area between Kharif and Rabi seasons using state-level agricultural inventory data while preventing any irrigation from taking place in June. However, this still fails to account for the spatial variability in the timing and duration of growing seasons across the country. This chapter describes the integration of the land use/land cover change dataset developed in Chapter 3 with a state of the art land surface model, the Joint UK Land and Environment Simulator, to generate soil moisture values considering land use/land cover change. Simulated soil moisture values are then post-processed using the irrigated area dataset developed in Chapter 4, temporally disaggregated to account for the growing seasons of individual crops, in order to include the effects of irrigation on soil moisture. The resulting datasets were used by other members of the Hydroflux India consortium to investigate the impact of large-scale irrigation across northern India on the behaviour of the South Asian monsoon. Due to time constraints imposed by the Hydroflux India programme, the soil moisture datasets were only developed for the Ganga basin in northern India.

Table 5.1.: Literature review of methods for representing the effects of irrigation on soil moisture at regional scales.

Reference	Study region	Time period	Irrigated area	Irrigated soil moisture
Boucher et al. (2004)	Global	1990	Döll et al. (2000)	Evapotranspiration (ET) set to potential rate in irrigated areas
Douglas et al. (2006)	India	1999–2000	Ramankutty and Foley (1999), combined with seasonal, state-level agricultural statistics on rainfed, irrigated and fallow cropland	ET set to potential rate in irrigated areas
Haddeland et al. (2007)	North America, Asia	1700–1992	Ramankutty and Foley (1999) and Siebert et al. (2005), scaled with historical estimates of national level irrigated area from FAOSTAT (FAO, 2015b)	Modelled using Variable Infiltration Capacity (VIC) model (Liang et al., 1994) irrigation scheme (Haddeland et al., 2006b,a)
Lobell et al. (2006)	Global	50-year scenario	Loveland et al. (2000), assuming all cropland is irrigated	Irrigated area set to θ_s when leaf area index, from Bonan et al. (2002), is above zero
Lobell et al. (2009)	Global	1978–1999	Siebert et al. (2005)	$> 0.4\theta_s$
Saeed et al. (2009)	Indian peninsula	1986–1992	Siebert et al. (2005)	$0.75\theta_s$

Continued on next page

Table 5.1 – *Continued from previous page*

Reference	Study region	Time period	Irrigated area	Irrigated soil moisture
Puma and Cook (2010)	Global	1902–2000	Wisser et al. (2010)	Soil moisture calculated on the basis of monthly irrigation rates estimated using a water balance model (Wisser et al., 2010). Soil moisture in irrigated areas set to holding capacity dependent on field capacity and vegetation-dependent rooting depth. Water depth over rice paddy set to 50mm
Shukla et al. (2014)			Wisser et al. (2010)	See entry for Puma and Cook (2010)
Tuinenburg et al. (2014)	India	1990–2000	Siebert et al. (2005)	$0.9\theta_s$

5.2 Methodology

5.2.1 Joint-UK Land and Environment Simulator

The Joint UK Land and Environment Simulator (JULES) is a process-based land surface model developed by the UK Met Office to use within its Unified Model (MetUM), an Earth system model. Within the Unified Model the purpose of JULES is to provide the lower boundary condition for an atmospheric GCM (Best et al., 2011). It has been widely used in "offline" mode for hydrological modelling applications (e.g. Zulkafli, 2014; Tsarouchi, 2014). Indeed, the model was recently found to reproduce adequately various aspects of the hydrology of the Upper Ganga basin, a sub-catchment of the Ganga basin (Tsarouchi, 2014). The present study used JULES version 2.2. The model is fully described by Best et al. (2011) and Clark et al. (2011a); however, a brief description of the model is provided here for completeness.

The model represents the land surface in terms of nine different surface types. Of these, five are vegetated (broadleaf forest, needleleaf forest, C_3 grass, C_4, shrubs) and four are non-vegetated (urban, inland water, bare soil and ice). Each surface type is associated with a unique parameter set describing its physical properties. Sub-grid heterogeneity of the land surface is characterised using the tiling method (e.g. Essery and Clark, 2003), in which each grid square is divided into user-specified fractions of the various surface types with discrete solutions of the surface energy balance equation. Overall fluxes of moisture, heat and momentum at the grid square level are the area-weighted averages of the respective fluxes calculated for each surface type (Best et al., 2011).

In offline mode JULES is supplied with meteorological forcing data including precipitation, radiation, air temperature, surface pressure, specific humidity and wind speed data. Input data files must have the same spatial extent and resolution as the model grid. For vegetated surfaces the amount of precipitation reaching the ground, or throughfall, is proportional to the fraction of the total canopy storage held by the vegetation. For non-vegetated surfaces, or when the canopy storage is full, throughfall is equal to precipitation. Water reaching the ground surface is partitioned into infiltration and surface runoff based on infiltration excess overland flow with vegetation-specific adjustments to account for soil macroporosity (Zulkafli et al., 2013). Potential evapotranspiration is calculated using the Penman-Monteith equation, extended to include a model of conductive heat transfer to the soil column (Cox et al., 1998). Canopy evaporation and evaporation from water bodies equals the potential rate, while transpiration from vegetated surfaces and bare soil evaporation are limited by stomatal resistance and the water content of the soil, respectively (Zulkafli et al., 2013).

Since JULES does not represent sub-grid heterogeneity in the subsurface, infiltration to the soil column is the area-weighted average of the infiltration calculated for the respective surface types. The soil column is divided into layers according to a user-specified discretisation scheme. Vertical groundwater flow is modelled using a finite difference approximation of the Darcy-Richards equation (Richards, 1931). The lower boundary condition is gravity drainage from the base of the soil column, which contributes to subsurface runoff (Best et al., 2011). The hydraulic characteristics of the soil, in terms of the relationship between soil moisture, pressure head and hydraulic conductivity, are defined using either the Brooks and Corey (1964) or Van Genuchten (1980) models. Root water uptake from subsurface layers is a function of root density, which varies exponentially with depth, and soil moisture stress.

5.2.2 Input data

Meteorological forcing data

Precipitation data was obtained from the daily gridded rainfall product of the Indian Meteorological Department (IMD) (Rajeevan et al., 2006). This dataset, which has a spatial resolution of 1 degree × 1 degree and provides daily rainfall data for the period 1951–2004, was created by interpolating daily observed rainfall data from 1,803 rain gauges distributed throughout India using inverse distance weighting (Rajeevan et al., 2006). The remaining meteorological input data, including radiation, air temperature, surface pressure, specific humidity and wind speed, were obtained from the NCEP reanalysis dataset (Sheffield et al., 2006); developed using multiple observation-based datasets from around the globe to correct systematic biases with the NCEP-NCAR reanalysis product. It has a spatial resolution of 1 degree × 1 degree and a 3-hourly, daily and monthly time step for the period 1948–2008.

Ancillary data

Land use/land cover data was obtained from the dataset developed in Chapter 3. This dataset was created using the Change in Land Use and its Effects (CLUE) model (Veldkamp and Fresco, 1996b; Verburg et al., 1999) to spatially disaggregate annual, district-level land use/land cover inventory data. Similarly, maps of irrigated area were obtained from the dataset presented in Chapter 4, developed by spatially allocating district-level agricultural data about the irrigated area of various crops according to maps of cropland extent and crop suitability. The land use/land cover dataset and the irrigated area dataset have a spatial resolution of 5 arcminutes × 5 arcminutes and provide maps on an annual basis between 1956–2010. Crop calendars specifying the growing season of irrigated crops in India

were obtained from the MIRCA2000 dataset (Portmann et al., 2010) and the AQUASTAT database (FAO, 2015a). Soil data for the study region was obtained from the UK Met Office Unified Model Central Ancillary Program (UM-CAP) at a spatial resolution of 0.1 degrees × 0.1 degrees. This dataset is based on soil texture maps from the Harmonised World Soil Database (FAO/IIASA/ISRIC/ISS-CAS/JRC, 2012) and ISRIC World Soil Information.

5.2.3 Land surface modelling

The model grid was specified for the Ganga basin with a spatial resolution of 0.1 degree × 0.1 degree, as shown in Figure 5.2. Input data were resampled to the model resolution using bilinear interpolation. Precipitation and meteorological forcing data were not downscaled from the original spatial resolution of 1 degree × 1 degree. The land use/land cover dataset was reclassified to the JULES land cover classification, as shown in Table 5.2. Figure 5.3, showing land use/land cover in the Ganga basin for the year 2000, highlights the fact that the study region is dominated by cropland. Based on previous research showing that, in India, JULES is insensitive to the distribution of broadleaf and needleleaf forest (Tsarouchi, 2014), the forest class from the land use/land cover change dataset was divided evenly between the two types. Cropland, which is not explicitly represented in the version of JULES [1] (Best et al., 2011), was defined as C_3 grass, following Van den Hoof et al. (2011) and Tsarouchi (2014), amongst others. The fallow class from the land use/land cover dataset was assigned to bare soil class in JULES.Default parameter values for vegetated and non-vegetated surface types, described by Best et al. (2011), were used throughout the analysis. This approach has previously been shown to produce satisfactory results for the Upper Ganga basin (Tsarouchi, 2014). The Brooks and Corey (1964) soil water retention model was used with spatially varying parameters provided by UM-CAP soil data.

The JULES simulation proceeded using an hourly time step between 1971–2005 using daily precipitation and meteorological data. This was to preserve the numerical stability of the model rather than to reflect the characteristics of the forcing data. Temporal downscaling of daily forcing data was carried out within JULES by assuming the values to be constant over the course of a day if the values represented the daily average (e.g. temperature) or by dividing by 24 if the values were cumulative (e.g. precipitation). A model spin-up was performed to initialise the model state. The model was considered to be properly initialised when the values of soil moisture and soil temperature at the end of

[1] JULES has included an explicit representation of agricultural land covers since version 4.0, which was released in early 2015. However, this was not available at the time the work presented in this chapter was carried out

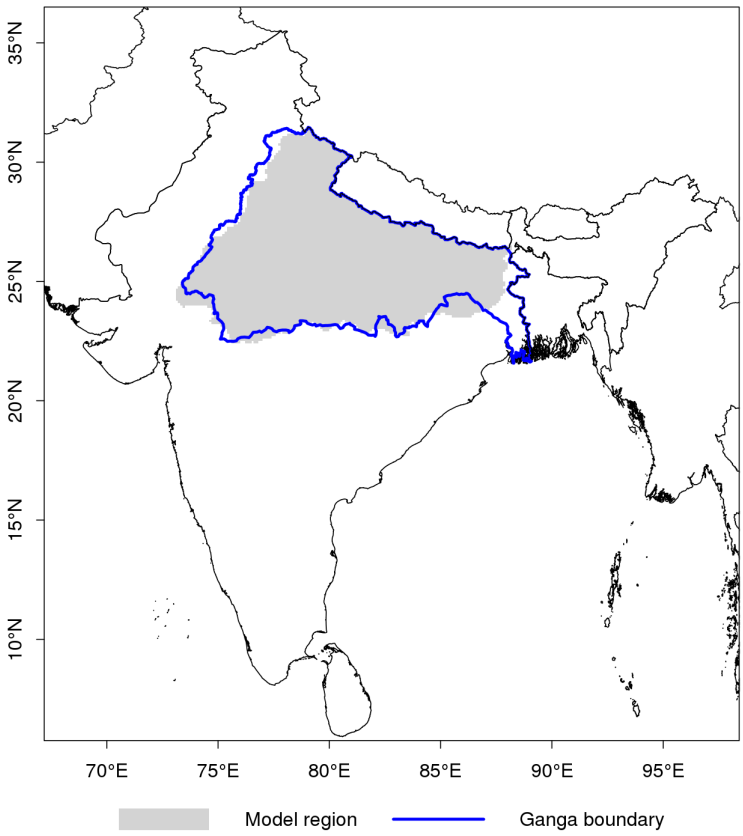


Figure 5.2.: Study region for land surface modelling, which focused on the Ganga basin in northern India due to time pressures imposed by the Hydroflux India programme. The model grid was set up with a spatial resolution of 0.1 degree × 0.1 degree in order to reuse input data prepared by other members of the Hydroflux India consortium.

the spin-up cycle had changed by no more than $1 kg/m^2$ and 1%, respectively, compared to the values at the end of the previous spin-up cycle. Model output was written to file with a daily time step. At the end of each year the model was paused to update the land cover information for the subsequent year. This was achieved by writing the internal model state to file at the end of each year, updating the control file to specify the temporal grid and land cover data for the following year, and restarting the model using the internal model state at the end of the previous year to initialise the model.

5.2.4 Post-processing

The irrigated area dataset was temporally disaggregated to a monthly time step using crop calendars from the MIRCA2000 dataset (Portmann et al., 2010) and AQUASTAT (FAO,

Table 5.2.: Translation of land use/land cover classification scheme to JULES land cover classes.

JULES	Land use/land cover (Chapter 3)
Broadleaf forest	50% Forest
	50% Tree crops
Needleleaf forest	50% Forest
	50% Tree crops
C_3 grass	Cropland
C_4 grass	Grassland
Shrubs	Shrubland
Urban	Urban
Inland water	Water
Bare soil	Barren/sparsely vegetated land
	Fallow
Ice	Permanent snow and ice

2015a). These calendars provide the start and end month of the growing season of various crops for four regions of India. For a given crop, the values in the annual irrigated area map are assigned to monthly maps corresponding to the respective growing season. At each time point the monthly irrigated area maps for the various crops are added together in order to produce a spatio-temporal dataset of the area actually irrigated on a monthly basis during the study region. Irrigated area values in each map were restricted to the area of cropland in the corresponding land use/land cover map.

Daily soil moisture values simulated by JULES were retrieved from the model output files. To account for irrigation it was assumed that the soil moisture in irrigated areas was kept at or above field capacity for the duration of the growing season. Field capacity is the maximum amount of water that can be held by the soil against the force of gravity and provides the ideal conditions for crop growth (Brouwer et al., 1985). It is usually taken as the water content at a pressure of -33kPa (Vanderlinden and Giráldez, 2011). For each grid square in the model region the field capacity of each layer in the soil profile was calculated using the Brooks and Corey model, expressed as

$$\frac{\theta}{\theta_s} = \left(\frac{\Psi}{\Psi_s}\right)^{-\frac{1}{b}} \tag{5.7}$$

where θ is water content, θ_s is saturated water content, Ψ is pressure head, Ψ_s is saturated pressure head and b is a model parameter. To obtain the field capacity of each grid square in the model region Eq. 5.7 was solved for θ, setting Ψ to -33kPa and using values of θ_s, Ψs and b from the UM-CAP soil input data:

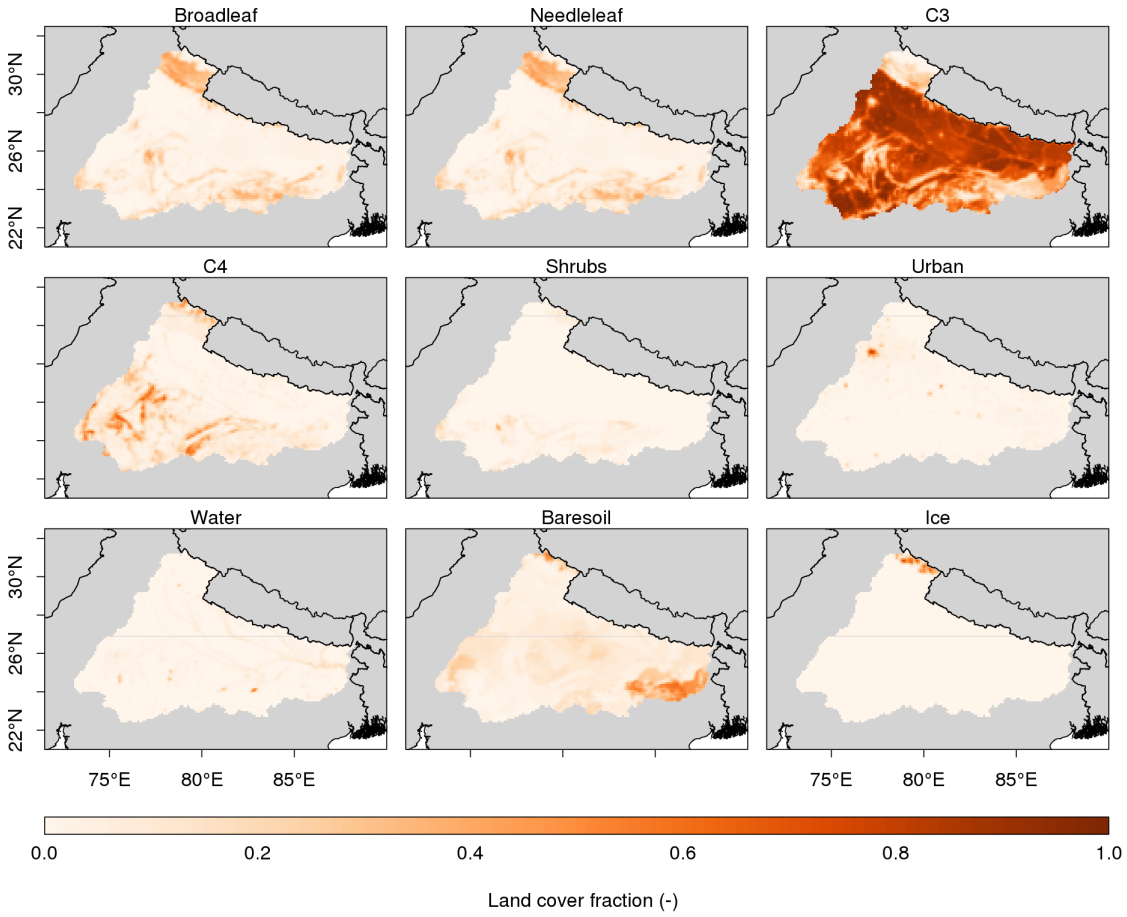


Figure 5.3.: Land cover in the Ganga basin for the year 2000. This data was derived from the land use/land cover dataset developed in Chapter 3 by reclassifying the original maps according to the scheme shown in Table 5.2 and resampling the resulting land cover maps to the model grid using bilinear interpolation.

$$\theta_{fc,i,z} = \theta_{s,i,z} \left(\frac{-33kPa}{\Psi_{s,i,z}} \right)^{-\frac{1}{b_{i,z}}} \tag{5.8}$$

where $\theta_{fc,i,z}$ is the field capacity in grid square i for soil layer z. In each grid square the daily soil moisture values simulated by JULES were compared to the field capacity to calculate the soil moisture deficit. If the simulated soil moisture exceeded field capacity then it was left unchanged. Otherwise, the value was increased by a specified amount in order to bring the irrigated fraction of the grid square, obtained from the monthly irrigated

area dataset, to field capacity, as follows:

$$\theta_{mod,i,t,z} = \theta_{jules,i,t,z} + (\theta_{fc,i,z} - \theta_{jules,i,t,z}) * irri_{i,t} \tag{5.9}$$

where θ_{mod} is the modified soil moisture for grid square i, time t and soil layer z, θ_{jules} is the modelled soil moisture and $irri$ is the total irrigated fraction.

5.3 Results

Figure 5.4 shows the land cover change scenario for the Ganga basin, demonstrating that land cover change has been relatively insubstantial during the period 1971–2004. In the same period, however, net irrigated area (the area irrigated at least once during a given year) has risen dramatically, as shown in Figure 5.5. The growth in irrigated area has mainly occurred in western parts of the Ganga basin, as shown in Figure 5.6. Annual irrigated maps were temporally disaggregated to monthly temporal resolution. Figure 5.7 shows the outcome of this procedure for 12 months between June 2000, the month when the Kharif (monsoon) season is generally considered to start, and May 2001. Irrigation in the Ganga basin was represented by modifying simulated soil moisture values in irrigated areas to minimally equal field capacity. Figure 5.8 shows the mean difference between the soil moisture values modified to account for irrigation and the values simulated by JULES for the same months shown in Figure 5.7. Meanwhile, Figure 5.9 shows the impact of irrigation on soil moisture during the Kharif and Rabi growing seasons over the course of the study period. It shows that Kharif irrigation, which supplements monsoon rainfall, is less intense than Rabi irrigation, which takes place during the dry winter months. This is further demonstrated in Figure 5.10, which shows the time series of the total volume of water in the top 0.1m soil layer across the Ganga basin with and without irrigation.

5.4 Discussion

The two soil moisture datasets provide daily maps with a spatial resolution of 0.1 degree × 0.1 degree showing the volumetric soil moisture for the period 1971–2005. The first dataset, the output of an offline JULES simulation, accounts for the effects of land use/land cover change but not irrigation, while the second version includes a simple representation of irrigation based on the assumption that the soil moisture of irrigated areas does not fall below field capacity during the growing season. This is considered to be a reasonable assumption for the Indo-Gangetic plain where almost all farmers have access to canal water, groundwater or both (Siebert et al., 2015). In the context of Hydroflux India the datasets were used in various experiments to supply the surface boundary condition of an atmospheric general

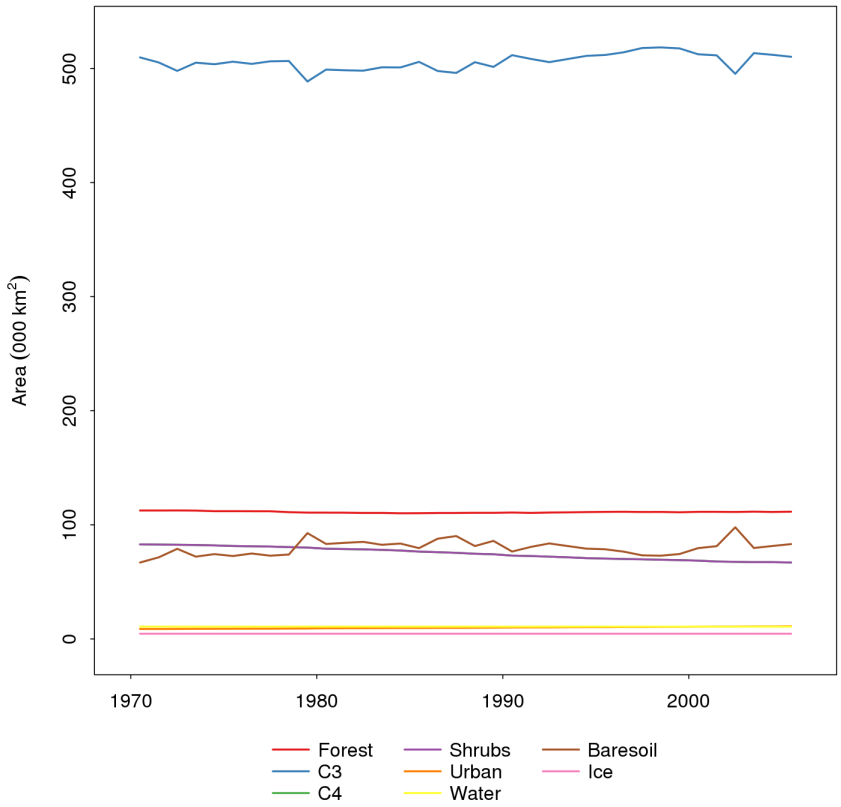


Figure 5.4.: Land cover change in the Ganga basin, 1971–2004.

circulation model (AGCM) to investigate the impact of large-scale environmental change on the behaviour of the South Asian monsoon. Initial results from one of these experiments, which is being carried out by researchers at IISc Bangalore, suggest that increased soil moisture at the land surface weakens the monsoon because increased soil moisture at the land surface as a result of irrigation increases latent heating and, as a result, decreases the surface air temperature. This has the effect of reducing atmospheric instability (the extent to which the atmosphere enhances convection) and therefore reduces the intensity of the monsoon circulation (Arindam Chakraborty, personal communication, 18 July 2014). However, these initial results should be treated with caution because detailed statistical analysis of the model output has not yet been carried out.

The methodology used in the present study improves upon previous attempts to represent irrigation processes in Earth system models in various ways. Firstly, JULES was run with a new land cover change dataset which provides a land cover map for each year of the model simulation and, furthermore, shows the proportion of cropland that is left fallow in any given year. Interannual variability and the impact of fallow land is not represented in

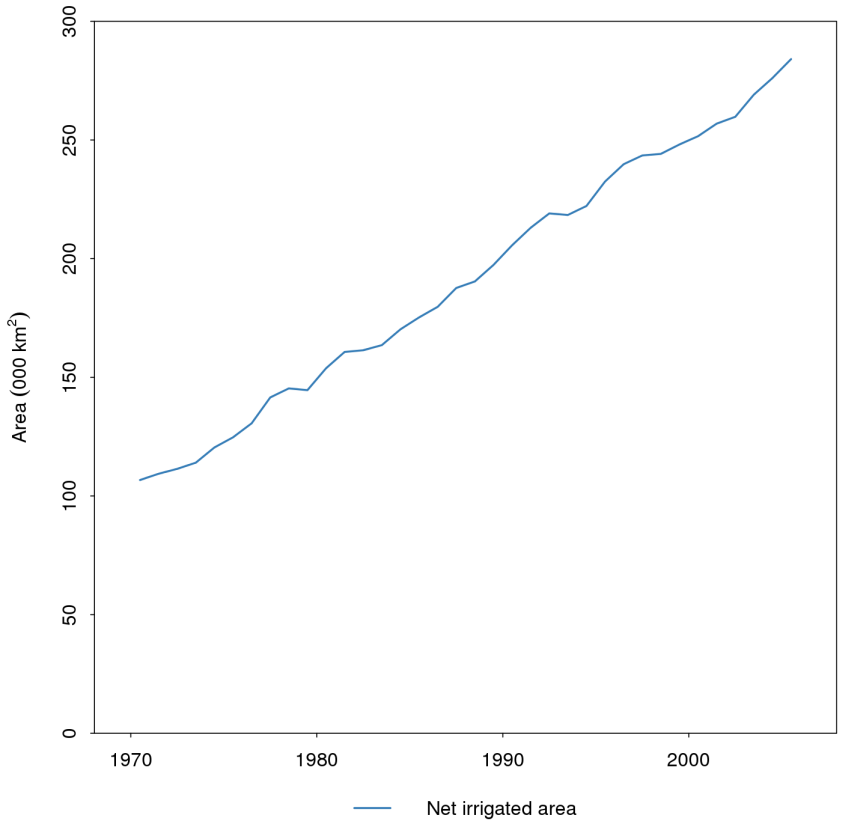


Figure 5.5.: Change in net irrigated area in the Ganga basin, 1971–2004.

alternative land cover change datasets (e.g. Ramankutty and Foley, 1999; Klein Goldewijk et al., 2011; Tian et al., 2014), which usually employ 5 or 10 year time steps. In order to represent the effects of irrigation on soil moisture the methodology incorporates a new irrigated area dataset, described in Chapter 4, based on district-level agricultural inventory data about the irrigated area of individual crops. This dataset also has an annual time step and shows the area actually irrigated rather than the area equipped for irrigation used by previous studies (e.g. Lobell et al., 2009; Saeed et al., 2009; Shukla et al., 2014). The annual irrigated area maps were further disaggregated to monthly maps using crop calendars obtained from MIRCA2000 (Portmann et al., 2010) and AQUASTAT (FAO, 2015a). As a result, the soil moisture dataset adjusted for irrigation incorporates the growing season of individual crops. This is important in India where positive soil moisture anomolies during boreal spring are correlated with decreased monsoon rainfall (Lee et al., 2009; Niyogi et al., 2010). Figure 5.10 shows the effect of this consideration on surface soil moisture. A weakness of the current study is the limited spatial and temporal extent of the JULES model setup, which resulted from time pressures imposed by the Hydroflux India programme. Thus, future work should run JULES with the land cover change dataset between 1956–

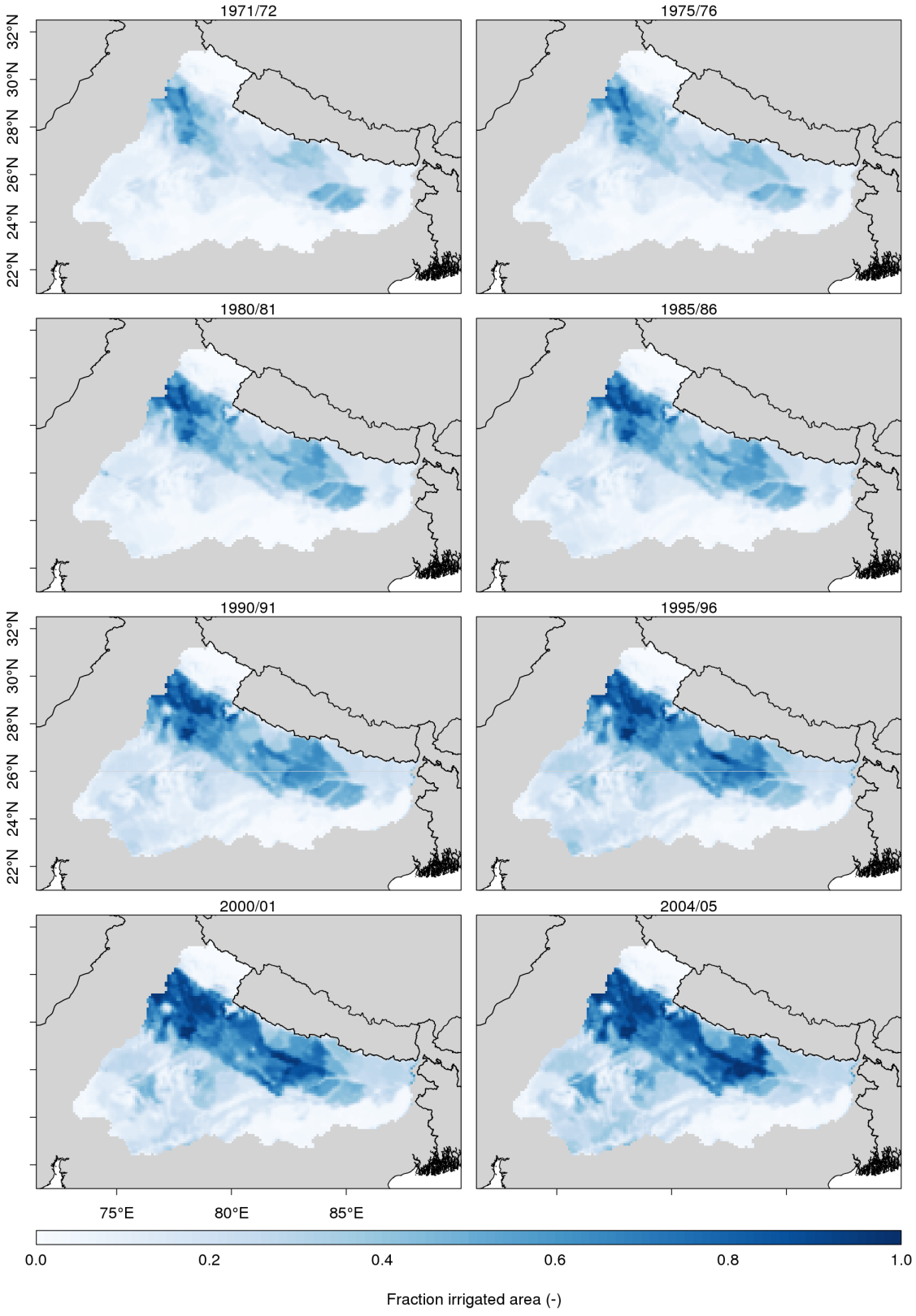


Figure 5.6.: Net irrigated area in the Ganga basin at 5 year intervals, 1971–2004.

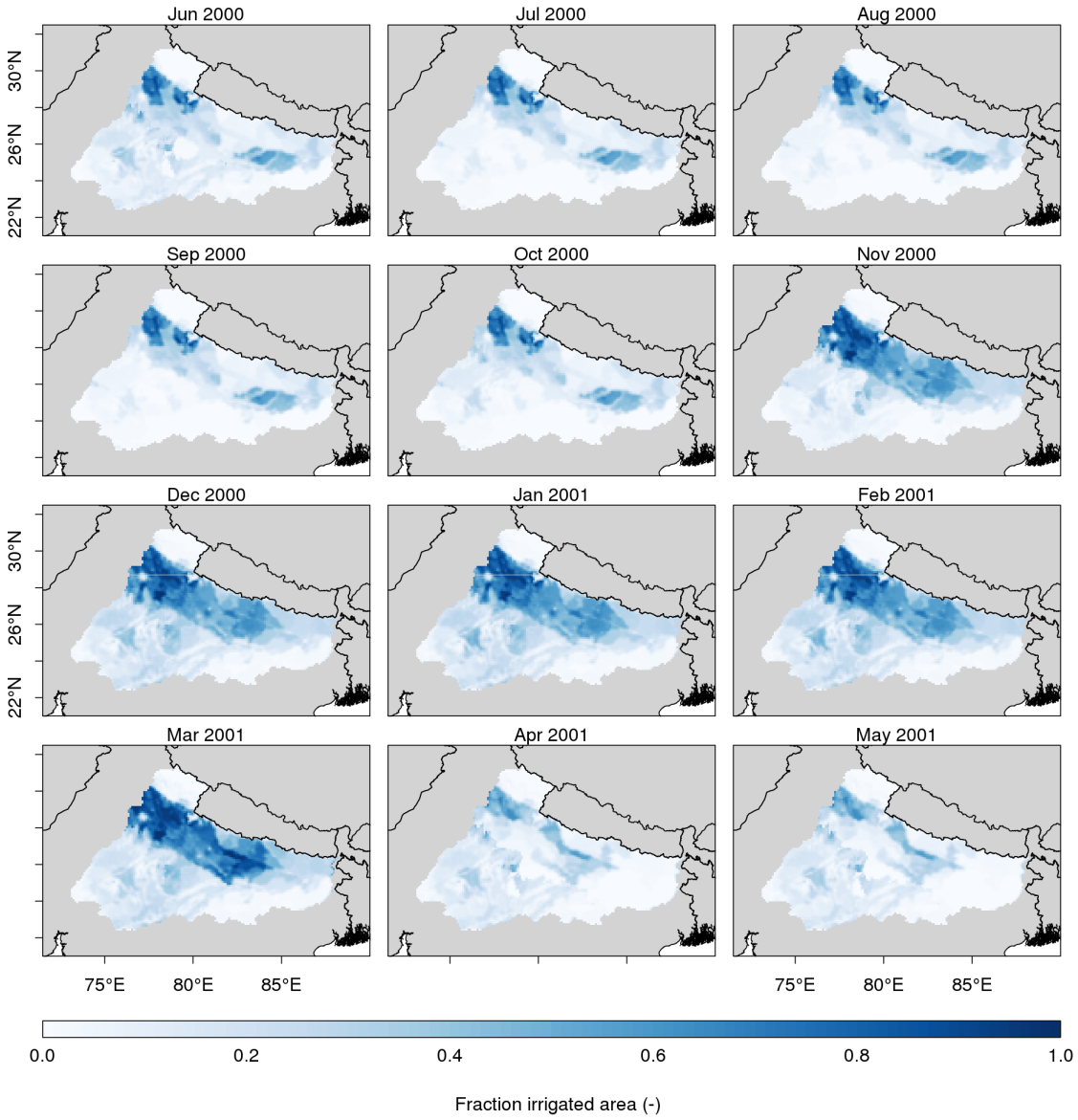


Figure 5.7.: Monthly irrigated area, June 2000–May 2001. In the Ganga basin, *Kharif* runs from June–October while *Rabi* is between November–March. Few crops are grown in the pre-monsoon season, April–May.

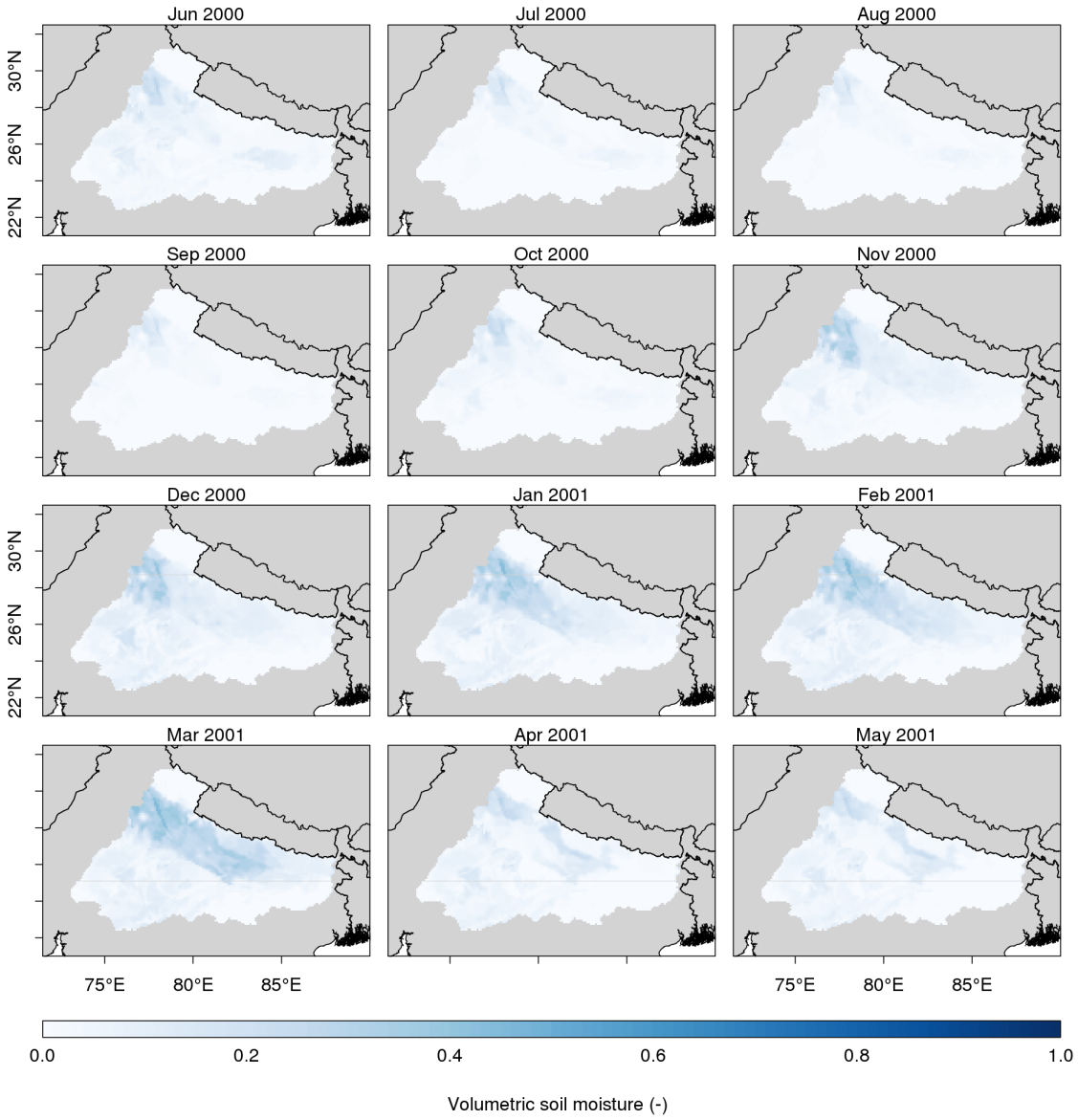


Figure 5.8.: Mean monthly difference between soil moisture values simulated by JULES and the corresponding values adjusted to include irrigation, June 2000–May 2001.

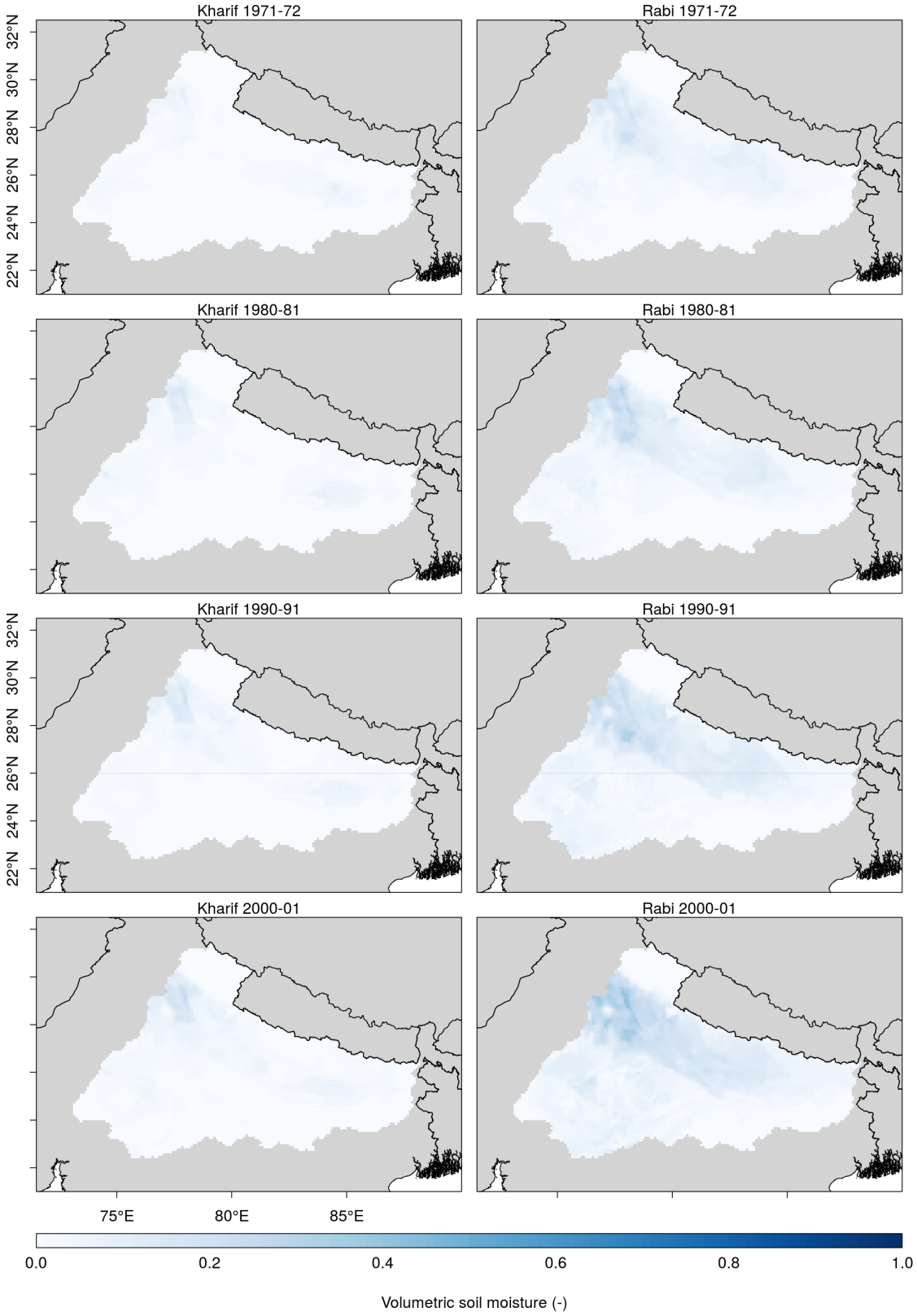


Figure 5.9.: Mean seasonal difference between soil moisture values simulated by JULES and the corresponding values adjusted for irrigation, at various time points during the study period.

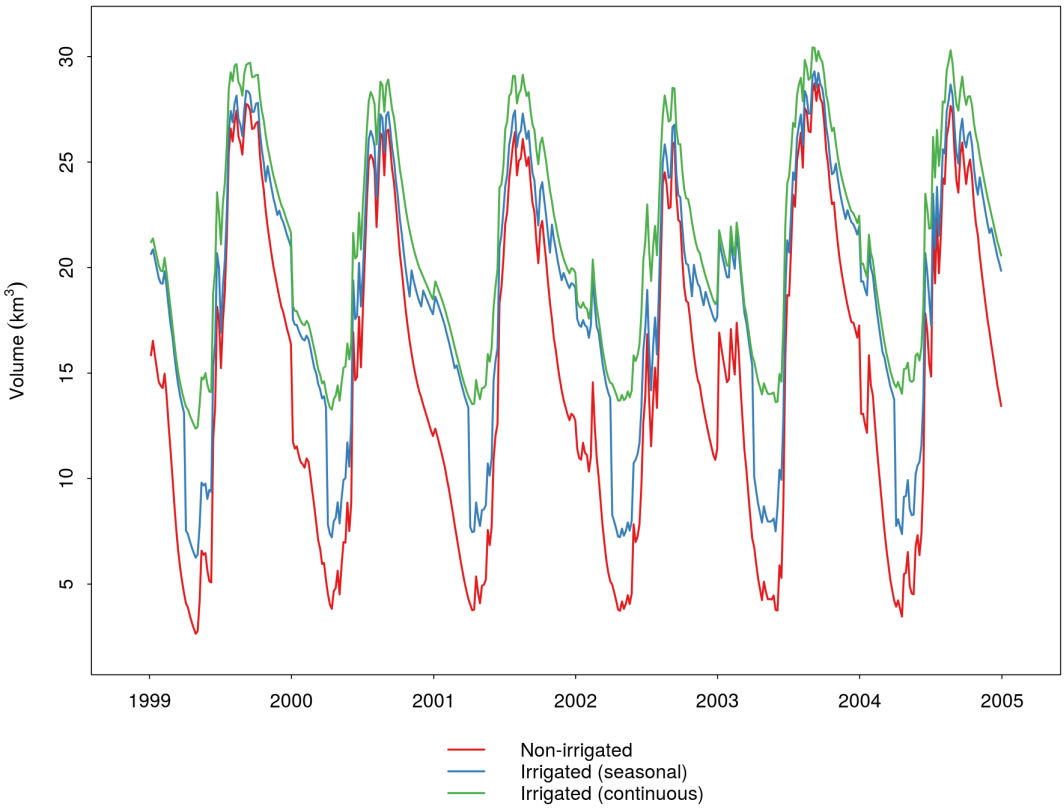


Figure 5.10.: Time series showing the total volume of water in the top 0.1m soil layer of the Ganga basin between 1999–2005. It again highlights that most irrigation occurs during the *Rabi* (November–March). The green line shows the irrigated volume if the growing seasons of individual crops are not taken into account.

2010, which properly captures the pre-green revolution period, and for the entire Indian subcontinent. This will allow the results of the climate model runs to be compared with previous work to quantify the effects of irrigation on the South Asian monsoon (e.g. Saeed et al., 2009; Puma and Cook, 2010; Shukla et al., 2014).

The soil moisture dataset is subject to various sources of uncertainty. Firstly, default parameter values for the various surface types represented in JULES are not optimised for Indian conditions. Process-based land surface models such as JULES require a large number of parameters which are very difficult to measure under field conditions (Beven and Cloke, 2012). Despite recent criticisms of process-based models as being over-parameterised and too complex given our lack of knowledge about the appropriate mathematical representation of hydrological processes at different spatial scales as well as the difficulties associated

with identifying the physical properties and boundary conditions of catchments (e.g. McDonnell et al., 2007; Beven and Cloke, 2012), Fatichi et al. (2016) argues that they remain the best tool for modelling the spatial distribution of soil moisture and modelling in data scarce regions. They are the only practicable option for modelling the Ganga basin given the unavailability of observed streamflow data with which to calibrate conceptual models. In the context of the present study, which aims to assess the sensitivity of a regional climate model to irrigated and non-irrigated conditions over the Indo-Gangetic Plain, the use of default parameters is considered a reasonable first approximation. However, it would be inappropriate to use the soil moisture datasets for any other purpose without first assessing the sensitivity of the model to different parameter settings. While it is unlikely that discharge data will become available in the near future, one option for validating JULES would be to compare the model output with evapotranspiration derived from remotely sensed imagery (Kunnath-Poovakka et al., 2016).

Additional uncertainty arises from the decision to force soil moisture in irrigated areas to remain at or above field capacity throughout the growing season. This approach relies on the assumption that water for irrigation is always available and that farmers will always irrigate if the soil moisture falls below the field capacity. In reality, however, water is not always available, especially if farmers rely on the supply-driven canal system for irrigation water. Nevertheless, across the Indo-Gangetic Plain many farmers who have access to canal water will also obtain water from groundwater (Thenkabail et al., 2005; Siebert et al., 2015). Moreover, water demand for irrigation varies during the growing season according to crop characteristics (Nazemi and Wheater, 2015a). Alternative methods for simulating irrigation estimate water demand for irrigation based on the difference between potential evapotranspiration, calculated separately for each crop, and the available crop water (Nazemi and Wheater, 2015a). However, the level of complexity required to accurately simulate these variables, especially the amount of water available to the crop at every time point, is currently beyond the capability of JULES. Moreover, the data requirements of such an approach are likely to be considerable. Furthermore, the use of a fixed crop calendar to specify the growing season of individual crops fails to account for interannual variability in the timing and duration of growing seasons (Nazemi and Wheater, 2015a). Lastly, uncertainty arises because of the coarse spatial resolution of precipitation and meteorological forcing data (1 degree $\times$ 1 degree) which fails to resolve the spatial variability of precipitation intensity in India (Pai et al., 2014). Furthermore, the use of daily precipitation data to force the land surface model means that the intensity of subdaily rainfall events is not properly represented. These two simplifications could result in errors in the simulated partition between surface runoff and infiltration (Singh, 2016). A sensitivity analysis by Tsarouchi (2014), modelling the Upper Ganga basin, revealed that the model is sensitive to alternative precipitation input datasets. Thus, further work is

needed to assess the potential impact of rainfall intensity on soil moisture.

The version of JULES (2.2) used here does not explicitly represent agricultural land cover types or management practices (Van den Hoof et al., 2011). Cropland is instead represented as C_3 grass, a natural vegetation type. This leads to inaccuracies in the representation of land surface processes because crops have substantially different physiological and structural properties in terms of albedo, leaf area, root depth, surface roughness and canopy height, among others (Van den Hoof et al., 2011). While the value of vegetation parameters are allowed to vary spatially and temporally (apart from root depth, which can only vary temporally), providing a more accurate representation of the crop growth cycle than using constant values, obtaining data to do this is challenging. One option is to supply leaf area index values derived from remote sensing, which are available from 1982 to the present day through the LAI3g dataset (Zhu et al., 2013). Alternatively, crop production models can be used to simulate dynamic vegetation growth and crop harvesting during a growing season (Nazemi and Wheater, 2015a). Dynamic coupling of land surface models with crop production models has been shown to improve the representation of land surface processes (e.g. Tsarouchi et al., 2014; Harding et al., 2015). However, to improve the quantification of land use/land cover change on surface fluxes, natural land cover and agricultural systems should be represented within a common conceptual framework (Bondeau et al., 2007; Van den Hoof et al., 2011; Osborne et al., 2015). Agricultural processes have been represented in JULES since the release of version 4.0 in early 2015 (Osborne et al., 2015). The inclusion of agricultural activities in land surface models will improve predictions about the impacts of climate change on food production and water resources (Osborne et al., 2015). Nevertheless, dynamic crop growth models will still be associated with uncertainty because, in reality, numerous factors influence farmer decision making that cannot be represented in macroscale modelling (Portmann et al., 2010; Nazemi and Wheater, 2015a; O'Keeffe et al., 2016).

The current workflow for representing human-environment interactions in process-based land surface models, whereby post-processing algorithms are applied to simulate the effects of human activities, is unsatisfactory. Instead, anthopogenic activities relating to water resources management should be included in Earth system models Nazemi and Wheater (2015a); Clark et al. (2015a). Several researchers have argued that the inclusion of irrigation activities is essential to properly represent the Indian climate. For example, (Saeed et al., 2009) argues that a warm bias over northern India observed in several regional climate models could be explained by the failure to account for processes associated with agricultural intensification, particularly irrigation. Meanwhile, Lobell et al. (2009) suggests that the failure to account for irrigation activities in Earth system models could explain temperature biases in models or conceal other sources of bias. Harding et al. (2015) point

out that to accurately model the effect of irrigation on precipitation it is necessary to simulate dynamic crop growth. Improving the ability of climate models to reliably simulate the South Asian monsoon is important to allow better prediction of intraseasonal monsoon variability (Turner and Annamalai, 2012). Krishnan et al. (2015) showed that forcing a global climate model with estimates of historical aerosol emissions and land use/land cover change, as well as greenhouse gas emission, improved its ability to simulate historical trends in mean monsoon rainfall. Similar work by Shukla et al. (2014) demonstrated that the inclusion of irrigation and greenhouse gas emissions in a coupled GCM is necessary to reproduce historical trends and variability of the South Asian monsoon over the last 50 years. The representation of irrigation and other human-water interactions in Earth system models is necessary to analyse the discrepancy between water supply and demand under climate change scenarios (Nazemi and Wheater, 2015a). Since version 4.1 JULES has included a simple irrigation module which increases soil moisture in irrigated areas to a critical point during the growing season of crops. The module allows the quantification of the effects of irrigation on soil moisture and surface fluxes but fails to consider the various impacts of irrigation on groundwater and surface water resources. This is necessary in Earth system models in order to analyse discrepancies between water supply and demand under climate change scenarios (Nazemi and Wheater, 2015a). Future model developments should therefore attempt to improve the representation of feedbacks between water supply and demand, particularly with respect to irrigation but eventually considering water demand from other sectors. In addition, there is an urgent need to add a groundwater component to JULES in order to properly capture interactions between groundwater and surface water. This is particularly important across the Indo-Gangetic plains where groundwater is relatively shallow in many places (MacDonald et al., 2016).

6 Hydrological model coupling and workflow orchestration in R

Chapters 3–5 describes a scientific workflow to reconstruct historical soil moisture values across the Ganges river basin. While the workflow was successful in the sense that the desired outcome was achieved, the procedure draws attention to the complexity of scientific workflows at regional scales and the challenges associated with processing and transforming heterogeneous datasets with different spatial and temporal characteristics into the information required to run hydrological models (Billah et al., 2016). It also highlights the inability of the current generation of spatially distributed, process-based hydrological models, including land surface models, to incorporate human-environment interactions. This chapter explores these issues further by setting up the R environment as a flexible workflow orchestration tool for hydrological data analysis and modelling. The suitability of R for this purpose is improved through the development of a new software package, *Hydro*, which provides a set of classes to represent multi-dimensional hydrological data and to form the basis of a common interface to hydrological models. The first section of this chapter reviews current trends in hydrological modelling and, in particular, draws attention to the aims of the current International Association of Hydrological Sciences Scientific Decade (2013-2022), "Panta Rhei–Everything Flows" (Montanari et al., 2013) and the emerging discipline of socio-hydrological modelling. Then, current technical solutions that address the challenges of building coupled modelling applications and scientific workflows are reviewed. The *Hydro* software is introduced and the experimental set-up is demonstrated through two example applications. Firstly, it is used to process meteorological time series data collected over a network of weather stations in the Plynlimon research catchments in Wales, United Kingdom. Then, it is used to implement a stylised reservoir operations model, originally developed by Garcia et al. (2016), which includes a proposed feedback mechanism between the reliability of water supply and population growth. On the basis of the example applications the main strengths and weaknesses of the software are discussed together with recommendations for future development.

6.1 Literature review

6.1.1 Current trends in hydrological modelling

Hydrological modelling is fundamental to the hydrological sciences (Buytaert et al., 2008). Models provide a representation of the hydrology of a catchment to forecast its response to future conditions or to simulate the behaviour of catchments where no output or state variables are measured (Beven, 2001). A useful classification scheme for hydrological models proposed by Wheater et al. (1993) divides models into empirical, conceptual and process-based models. Empirical models seek to characterise aspects of catchment behaviour from observations of input and output variables (Wheater, 2002; Pechlivanidis et al., 2011). Conceptual models consist of simplified representations of the hydrological processes perceived to influence the system response and include parameters that do not correspond with physical quantities and must therefore be inferred from observations (Kavetski et al., 2006). Process-based models are based on mathematical equations describing physical processes (Montanari and Koutsoyiannis, 2012; Beven and Young, 2013). They have variously been referred to as "physics-based" or "physically-based" models; however, since all models in practice contain simplifications of the actual physics, "process-based" is a more precise description of the modelling approach (Montanari and Koutsoyiannis, 2012). The International Association of Hydrological Sciences (IAHS) Scientific Decade (2003–2012) Predictions in Ungauged Basins (PUB) was formulated on the basis of concern amongst the scientific hydrological modelling community that empirical methods were unable to deal with a range of issues that required predictions of streamflow in ungauged basins or under non-stationary conditions (Sivapalan, 2003). Thus, over the course of the Scientific Decade there was a concerted effort to improve scientific understanding of catchment scale hydrological processes (Hrachowitz et al., 2013). Following the PUB initiative there is consensus in the scientific hydrological community that models should be viewed as hypotheses about system behaviour and that modelling itself is a learning process (Hrachowitz et al., 2013).

Hydrological models are associated with a large amount of epistemic uncertainty arising from a lack of knowledge about the appropriate mathematical representation of hydrological processes at different spatial scales, a lack of knowledge about the physical properties and boundary conditions of catchments and a lack of knowledge about the reliability of input and validation datasets (Beven, 1989; Beven and Cloke, 2012; Clark et al., 2011b, 2015a; Gupta et al., 2012; Mount et al., 2016). While in theory process-based models contain parameters that refer to measurable quantities, in practice models rely on calibrated, "effective" parameter values to implicitly account for the spatial heterogeneity of catchment properties and incomplete process understanding (McDonnell et al., 2007; Savenije, 2009; Sivakumar, 2008; Beven and Cloke, 2012). Thus, both conceptual and process-based models are calibrated to some degree giving rise to the notion that for a given place various

model structures and parameter sets are capable of simulating response variables equally well; a concept known as equifinality (Beven, 2001, 2006; Beven et al., 2012). Consequently, there is a lack of agreement about the appropriate model structure for different applications, which has led to the development of numerous hydrological models to address a wide range of scientific and engineering questions (Beven, 2001; Clark et al., 2011b; Weiler and Beven, 2015).

The various models accumulated by the hydrological modelling community may be viewed as working hypotheses about the hydrological behaviour of various places (Clark et al., 2008; Savenije, 2009; Clark et al., 2011b; Gupta and Nearing, 2014). To develop better models there is a need to identify the strengths and weaknesses of current models and highlight the processes and systems that are poorly understood in order to drive the collection of new or more accurate data and devise field studies of specific processes (Buytaert et al., 2008; Clark et al., 2012; Gupta et al., 2012). This is one of the main aims of uncertainty analysis (Buytaert et al., 2008; Clark et al., 2012; Gupta et al., 2012), and fundamental to the notion of modelling as a learning process. However, while there have been several model intercomparison studies in recent years (e.g. Wood et al., 1998; Yang et al., 2000; Reed et al., 2004; Pebesma and Bivand, 2005; Duan et al., 2006; Breuer et al., 2009; Smith et al., 2012), these have largely failed to diagnose differences in model performance because the structure and implementation of different models varies so widely (Kampf and Burges, 2007; Fenicia et al., 2008; Clark et al., 2011b, 2015a). The problem is made worse by the fact that it is hard to resolve structural errors in calibrated models because of problems arising from equifinality (Kirchner, 2006).

According to Clark et al. (2011b), hydrological models should be viewed as a collection of coupled hypotheses about different aspects of the catchment system. These component hypotheses cannot necessarily be tested at the level of the system as a whole because of potentially complex relationships between constituent parts within the model structure that can obfuscate model deficiencies (Clark et al., 2011b, 2012). Thus, in order to test, and potentially reject, model structures as hypotheses it is necessary to isolate model components and evaluate them independently as well as in combination with different representations of linked processes (Clark et al., 2011b). This is challenging because existing models vary substantially in conceptualisation and implementation (Castronova et al., 2013; Clark et al., 2015a; Billah et al., 2016). The fact that some hydrological models are released as closed source software packages is a fundamental impediment to scientific progress which limits the reproducibility of scientific results and forces modellers to duplicate the work of others (Morin et al., 2012; Steiniger and Hunter, 2013; Moulds et al., 2015). As Buytaert et al. (2008) point out, many implementations of hydrological models, whether open or closed source, follow rigid structures which restrict access to internal state variables and make it

difficult for modellers to adapt or improve the model code for their own purposes. This is exacerbated by poor programming practices amongst model developers, perhaps arising from the fact that many geoscientists have received no formal training in programming (Joppa et al., 2013; Wilson et al., 2014), which can also make it difficult to even understand the implementation of model algorithms. As a result of these issues, model development to date has occurred on an *ad hoc* basis, relying on the efforts of individuals and research groups rather than a systematic effort by the entire hydrological modelling community (Argent et al., 2006; Buytaert et al., 2008; Clark et al., 2015a; Troy et al., 2015; Weiler and Beven, 2015; Clark et al., 2016). Moreover, a lack of discourse between modellers and field experimentalists means that the current generation of hydrological models does not always reflect advances in process understanding gained from field studies (McDonnell et al., 2007; Clark et al., 2011b).

Recognising these issues, members of the community have started to question whether a new approach to hydrological modelling is required (e.g. Sivapalan, 2009; Savenije, 2009; Weiler and Beven, 2015). In particular, there has been a concerted effort to develop flexible modelling frameworks that facilitate the systematic development, evaluation and comparison of multiple model structures as hypotheses of catchment behaviour, and the inclusion of additional and alternative process representations as our perceptual understanding of the system improves. This approach recognises the fact that each catchment is unique and that fixed model structures are incapable of handling natural variability in space and time (Beven, 2000). In conceptual modelling the Framework for Understanding Structural Errors (FUSE) (Clark et al., 2008) allows different representations of model components within a fixed model architecture, while SUPERFLEX (Fenicia et al., 2011; Kavetski and Fenicia, 2011) provides a more flexible approach to model building by allowing various combinations of generic model components. The Structure for Unifying Multiple Modeling Alternatives (SUMMA) (Clark et al., 2015b,c) aims to provide a unified framework for developing process-based hydrological models. Essentially, SUMMA implements a set of general mass and energy conservation equations with multiple options for spatially discretising the model domain and various alternative flux parameterisations for representing sub-grid spatial heterogeneity (Clark et al., 2015b). These frameworks are implemented as open source software projects with core developers encouraging users of the software to contribute to model development (e.g. Clark et al., 2015b). While they are useful modelling tools in themselves, they may also be used as test-beds for a community hydrological model of the type proposed by Weiler and Beven (2015) or the next generation of "hyper-resolution" land surface models suggested by Wood et al. (2011) and debated elsewhere (Beven and Cloke, 2012; Wood et al., 2012; Nazemi and Wheater, 2015a). In this way, modelling frameworks facilitate collaboration between different parts of the hydrological modelling community in order to advance the theoretical basis of hydrological models and

address major research challenges (Archfield et al., 2015; Clark et al., 2016).

A recent advance in our perceptual understanding of hydrological systems is the recognition that the water cycle is no longer governed entirely by natural processes but also, increasingly, by anthropogenic activities (Montanari et al., 2013; Savenije et al., 2014; Sivapalan and Blöschl, 2015). This has given rise to the nascent field of socio-hydrology, which is concerned with understanding the feedbacks and interactions between coupled human-water systems in order to improve the resilience and sustainability of these systems under environmental and social change (Sivapalan et al., 2012). Its fundamental importance to the wider discipline of hydrology is demonstrated by its central role in defining the current IAHS Scientific Decade (2013-2022), "Panta Rhei–Everything Flows" (Montanari et al., 2013), which aims to gain new insight into the processes governing the hydrological cycle by concentrating research activities on the interactions between society and the environment (Montanari et al., 2013; McMillan et al., 2016). In their introduction to the Scientific Decade, Montanari et al. (2013) draw attention to the fact that the current generation of hydrological models have mostly been developed for the analysis of pristine catchments with human impacts on the environment modelled separately. Socio-hydrological modelling instead treats anthropogenic activities as an integral part of the water cycle rather than an external forcing (Troy et al., 2015). It aims to provide insight into the mechanisms and drivers of human-water interactions, support decision-making and policy development and make forecasts and predictions about the response of socio-hydrological systems under future conditions (Kelly (Letcher) et al., 2013; Blair and Buytaert, 2016).

Coupled human-water systems are complex systems involving 'multiple interacting components, local connections and nonlinear relationships between the components' (Troy et al., 2015). Developing models of these systems is challenging for various reasons. This includes the fact that different system components interact at different spatial and temporal scales (Elshafei et al., 2014). There is epistemic uncertainty about the relationships between system components, particularly concerning indirect relationships, alongside natural variability and epistemic uncertainty associated with models of individual system components, including those representing hydrological processes (Elshafei et al., 2014; Di Baldassarre et al., 2015; Mount et al., 2016). Furthermore, it requires effective collaboration between diverse research communities in order to include the state of the art in terms of the knowledge and understanding of system components (Blair and Buytaert, 2016), which is often difficult (Montanari et al., 2013). Current socio-hydrological models mainly consist of stylized relationships intended to develop hypotheses about the system and explore its behaviour under different scenarios of social and environmental change (e.g. van Emmerik et al., 2014; Di Baldassarre et al., 2015; Garcia et al., 2016). However, in the future there will be a need to incorporate more realistic models capable of providing insight into socio-

hydrological systems rather than simply reinforcing what is already known (Troy et al., 2015; Loucks, 2015; Blair and Buytaert, 2016), as well as making predictions about the co-evolution of social and hydrological systems under future conditions. This will involve more complete models of human behaviour (Loucks, 2015), as well as more accurate models of hydrological systems (Gober and Wheater, 2015; Mount et al., 2016).

6.1.2 Modelling complex systems

Modelling complex hydrological systems requires the integration of heterogeneous data and models from various sources (Billah et al., 2016). Scientific workflows express a sequence of tasks required to process, transform and analyse data to achieve a desired scientific outcome (Gil et al., 2007; McPhillips et al., 2009; Billah et al., 2016). Thus, alongside the scientific challenge of designing scientific workflows to answer specific questions is the technical challenge of implementing them efficiently and reproducibly. Model coupling has emerged as an effective way to build integrated environmental models (Goodall et al., 2011; Bulatewicz et al., 2013; Yue et al., 2015). In recent years several tools have been developed to facilitate model coupling based on the notion of component-based frameworks (Argent, 2004). These tools provide a standard interface to facilitate interaction between model components (Valcke et al., 2012). The need for couplers was addressed by the climate modelling community to couple separate components of Earth system models, resulting in solutions including the Earth Systems Modeling Framework (Hill et al., 2004), Ocean Atmosphere Sea Ice Soil (OASIS), and the Community Surface Dynamics Modeling System (CSDMS) (Syvitski et al., 2004), which are designed to couple the core components of Earth system models. There are few applications of these technologies to hydrological modelling, although Shrestha et al. (2014) coupled a land surface model (NCAR Community Land Model version 3.5), a groundwater model (ParFlow) and an atmospheric model (Consortium for Small-Scale Modeling, COSMO), using OASIS version 3.0 to create a fully integrated Earth system model including groundwater dynamics, which most Earth system models ignore (Maxwell and Miller, 2005). Coupling technologies for Earth system modelling result in tightly coupled applications in which the source code of the respective model components is ported to a single modelling application (Goodall et al., 2011).

In the context of environmental modelling frameworks, invasiveness is defined as 'the degree to which model code is coupled to the underlying framework' (Lloyd et al., 2011). Invasiveness occurs for various reasons including the provision of an Application Programming Interface (API) and data structures for which model developers must provide interfaces to leverage framework services, the need for boilerplate or "non-science" code to enable a model to run under a particular framework, and specific implementation requirements of the framework related to programming languages, operating systems and the availability

of specific software libraries (Lloyd et al., 2011). In Earth system modelling, where computational efficiency is paramount, a certain level of invasiveness is necessary to achieve the most efficient workflow in terms of process representation and data exchange between components (Goodall et al., 2011; Buahin and Horsburgh, 2015). In other modelling applications, however, framework invasiveness has several drawbacks. Firstly, modifying the source code of models is time consuming and requires a high level of ability in various programming languages and techniques (Bulatewicz et al., 2013; David et al., 2013; Dozier et al., 2016). This may be acceptable if the structure of the coupling application is known at the outset and involves a small number of model components. In practice, however, scientific workflows are refined iteratively and may change considerably as hypotheses are rejected and new ones are developed (McGuire et al., 2007; Dunn et al., 2008; Fenicia et al., 2008; Shao et al., 2009; Görlach et al., 2011). Adding a large amount of additional code to the model implementation increases the complexity of the software by forcing it to follow the conventions of the framework rather than following the most efficient structure for the model component itself. This can result in obfuscated model code that is difficult to understand sufficiently to fix bugs or make improvements. It also increases the likelihood of introducing errors to the model code (Lloyd et al., 2011; David et al., 2013). Invasive modelling frameworks affect the portability of models because they limit the extent to which models can be used independently or within alternative model framework (Lloyd et al., 2011). Moreover, adapting a model to a particular framework, unless it is performed by the model developers themselves, will remove the model implementation from its development cycle (Voinov and Shugart, 2013; David et al., 2013).

The Open Modeling Interface (OpenMI) Standard (Moore and Tindall, 2005; Gregersen et al., 2007) is a 'software component interface definition' designed to promote interoperability between new and existing models (Moore and Tindall, 2005; Gregersen et al., 2007; Goodall et al., 2011). It follows a loose integration approach in which model components adopt a standard interface allowing them to exchange data at runtime (Moore and Tindall, 2005). This gives developers of model components greater freedom to implement model equations in the most efficient way possible, while developers of coupled system models have greater flexibility about the model structure (Goodall et al., 2011). The OpenMI Standard is based on the principles of object-oriented programming and defined in terms of the operations a class must implement (Bulatewicz et al., 2013). The OpenMI Software Development Kit (SDK) is a modelling framework implementing the OpenMI Standard that provides the core functionality required to link model components as well as helper functions for data management (Castronova and Goodall, 2013; Knapen et al., 2013). Both the Standard and the SDK are implemented in the object-oriented languages C# and Java (Bulatewicz et al., 2013). Thus, to adapt an existing model to the OpenMI Standard it is necessary to represent the model as a C# or Java class implementing a set of de-

fined operations (Bulatewicz et al., 2013). The model source code must be structured as three functions called Initialize, Perform Time Step and Finish (Castronova and Goodall, 2013). An additional 15 helper functions must be defined to perform various tasks related to the execution of the model at runtime and the model description (Bulatewicz et al., 2013). This is relatively straightforward if the original model is written in a supported language but considerably more challenging otherwise. For this reason the framework is considered relatively invasive because the original model must be substantially modified in order to take advantage of framework services (Knapen et al., 2013). The OpenMI Standard is designed primarily for Windows and support for other operating systems is variable. It has been ported to Linux using the Mono compiler although according to the project website it is only tested on one Linux distribution (OpenSUSE 11.0, 64-bit) and it is unclear whether the development cycle for the Linux version follows that of the main Windows version. A further weakness of OpenMI is that it is focused on the hydrology domain. Indeed, Knapen et al. (2013) highlight the fact that the OpenMI Standard is not well adapted to models other than hydrological models. This is a major weakness of the OpenMI approach considering the emerging trends in hydrological modelling towards a more holistic, interdisciplinary treatment of the hydrological cycle (Montanari et al., 2013).

The Object Modeling System (OMS) (David et al., 2013) is an alternative, Java-based framework for environmental modelling that is less invasive than OpenMI (David et al., 2013; Formetta et al., 2014). It was originally conceived as an evolution of the Modular Modeling System (MMS) (Leavesley et al., 2006), a component-based implementation of the Precipitation-Runoff Modeling System (PRMS) (Leavesley et al., 1983). The current version of the software, OMS Version 3 (OMS3; David et al., 2010) substantially changed the design of previous versions to deliver a less intrusive, more flexible modelling framework. Where traditional modelling frameworks such as OpenMI and previous OMS versions provide an API with which components must interface, OMS3 is a lightweight framework based on the principle of "inversion of control", or the idea that the source code of components should not call framework APIs directly (Lloyd et al., 2011). Instead, OMS3 uses language annotations within the source code of model components to specify the point in the code where functionality from the modelling framework is required (David et al., 2010). Lightweight frameworks are generally easier to learn because prospective users do not require detailed understanding of the framework's API and data types (David et al., 2013). Nevertheless, OMS3 contains powerful features including sophisticated routines for automatic calibration and sensitivity analysis as well as supporting implicit parallelism (Formetta et al., 2016). Furthermore, it interacts directly with uDig-JGrass, an open-source GIS, in order to provide tools for handling spatially explicit input/output data and visualisation (Formetta et al., 2016). Models implemented in the framework include NewAge-JGrass (Formetta et al., 2014), AgES (Green et al., 2015), AgES-W (Ascough II

et al., 2012) and PRMS-OMS (David et al., 2002). In addition, Formetta et al. (2016) recently implemented the spatially distributed, process-based model GEOtop (Rigon et al., 2006), with a view to providing a template for the future integration of process-based models.

The process of setting up a workflow for hydrologial data analysis and processing can be considered as a type of service orchestration, which refers to the coordination and arrangement of multiple services to execute a service-oriented workflow (Barbosa and Barbosa, 2009; Bensmann et al., 2014). The OpenMI Configuration Editor is one solution to service orchestration which allows modellers to visually organise and execute scientific workflows involving coupled model components through a graphical user interface (Goodall et al., 2011; Bulatewicz et al., 2013). It was used by Goodall et al. (2011) to orchestrate a simple rainfall-runoff model consisting of three components to calculate excess rainfall, incremental runoff and river routing using implementations of the Curve Number method, Unit Hydrograph method and Muskingham Routing method, respectively, exposed as Web services. It was also used by Goodall et al. (2013) to coordinate a two-way coupling between the Community Atmosphere Model, implemented within the Earth System Modeling Framework, and the OpenMI-compliant Soil Water Assessment Tool (Betrie et al., 2011). Kepler (Altintas et al., 2004) is a scientific workflow system in which workflows are expressed as directed graphs, with nodes representing computational tasks and connections representing the flow of data between nodes (McPhillips et al., 2009; Goodall et al., 2011). In common with the OpenMI Configuration Editor it provides a graphical user interface which allows users to visually construct a scientific workflow. One of the main advantages of this approach is the fact that it allows workflows to be expressed clearly and in a way that makes it straightforward to see what a workflow will do when it is executed (McPhillips et al., 2009). Furthermore, it hides unnecessary implementation details from workflow developers and therefore encourages the reuse and adaptation of existing workflows (McPhillips et al., 2009). Meanwhile, OMS3 model applications are coordinated using a Domain Specific Language based on Groovy, an object-oriented scripting language for Java (David et al., 2010).

One drawback of specialist tools for service orchestration, such as those used by OpenMI and OMS3, is that they often lack the appropriate functionality to link models following different approaches or with different spatial and temporal resolutions (Ludwig, 2011). They are also strongly coupled to the underlying modelling framework and require model components to conform with the relevant standard. Scripting langauges, which are designed primarily for "gluing" together various functional components either written in the same language or in system programming languages such as C/C++ and Fortran (Ousterhout, 1998), provide a more flexible approach to service orchestration (Ousterhout, 1998; Bu et al., 2015). However, while the Groovy language for workflow orchestration using OMS3

components is more flexible than GUI-based approaches such as the OpenMI Configuration Editor or Kepler, it still lacks much of the functionality of data processing languages such as R and Python. These langauges provide sophisticated methods for handling and processing data in various forms, high-level functionality for accessing Web resources (Bu et al., 2015), and several tools for visualisation and data exploration.

This chapter describes the results of an experiment to set up R as a flexible and relatively easy to learn workflow orchestration tool for hydrological data analysis and modelling. To further enhance the utility of the R system for hydrological workflow orchestration a new software package, *Hydro*, is developed. A detailed desciption of the software package is provided in the next section, followed by an example application of R to orchestrate a simple workflow from the emergent discipline of socio-hydrology.

6.2 Software description

The aim of *Hydro* is to provide a set of classes that facilitate the use of the R system for orchestrating scientific workflows. To this end, the Hydro package implements a set of classes and methods designed to represent different types of hydrological data within the workflow and to form the basis of a common interface to hydrological models. The package is based on the S4 system for object-oriented programming (Chambers, 1998, 2008). This system is better suited to complex data structures compared with the alternative and more informal S3 system because objects are validated against a formal class definition when they are created. This minimally ensures that each component, or "slot" in S4 terminology, has the correct class but also allows additional validity criteria to be imposed through a validity function included with the class definition.

6.2.1 Classes

According to Chambers (1998), 'classes are the fundamental organising principle for data'. The *Hydro* package is built on three classes representing different space-time geometries: HydroSTF, HydroSTS and HydroSTI. Hereafter, these classes will be collectively referred to as HydroST* classes. These classes inherit from *spacetime* (Pebesma, 2012) classes STF, STS and STI, respectively, which all derive from base class ST and contain slots `sp`, to represent space, and `time` and `endTime` to define the start and end time, respectively, of each data point. HydroST* classes additionally include slots `z` and `endz`, analogous to `time` and `endTime` attributes in class ST, to define depth profiles. This is important for hydrological and meteorological variables measured and simulated at specific vertical levels above or below the Earth's surface, such as soil moisture and wind speed. Figure 6.1

shows the relationship between *spacetime* and *Hydro* classes and the methods defined for *spacetime* classes which are subsequently inherited by the HydroST* classes.

Classes derived from HydroSTF, HydroSTS and HydroSTI include a `data` slot to represent data in various formats. Objects belonging to these classes should represent a single hydrological variable. An essential feature of the HydroST* classes in the context of service orchestration is the inclusion of a slot `metadata` to provide an unambiguous description of the data using entries from the CUAHSI controlled vocabulary (Horsburgh et al., 2014). The class diagram for *Hydro* is shown in Figure 6.2.

Class HydroSTF represents the geometry of data with a full space-time grid, which means that for n spatial features, which may be points, lines, polygons or grid cells, and m time points, $n \times m$ data points are available (Pebesma, 2012). Three derived classes of HydroSTF include data in different formats:

HydroSTFDF objects store data as a `data.frame` in long format with columns representing depth profiles. This contrasts with STFDF objects from *spacetime* where columns are assumed to refer to separate variables.

HydroSTF.raster objects represent spatially gridded data stored as a RasterStack object from package *raster*. For a geometry with m time points and p depth profiles the RasterStack has $m \times p$ layers with depth cycling fastest. The number of spatial features equals the number of cells in the RasterStack object.

HydroSTF.array objects, which also represent spatially gridded data, store data as a four-dimensional array (time, x, y, z).

Of these derived classes HydroSTFDF is the most flexible because it can represent the most spatial feature types (points, lines, polygons and pixels). A typical use of this class would be to represent precipitation data collected over regular time intervals by a network of rain gauges. Classes HydroSTF.raster and HydroSTF.array should be used to represent spatially gridded data products such as those derived from remotely sensed data or the output of spatially distributed hydrological models.

Class HydroSTS represents data on a sparse space-time grid. This is similar to a full space-time grid except that only non-missing data is stored. In these objects, an index $[i,j]$ is associated with each data value giving the spatial feature i and time point j to which it belongs. This layout can be more efficient than a full space-time grid under various circumstances described by Pebesma (2012). Data is included as a `data.frame` in the derived class HydroSTSDF. Data suited to this layout include the activation of flood defence systems or rainfall data in regions with extended dry spells. While the first two

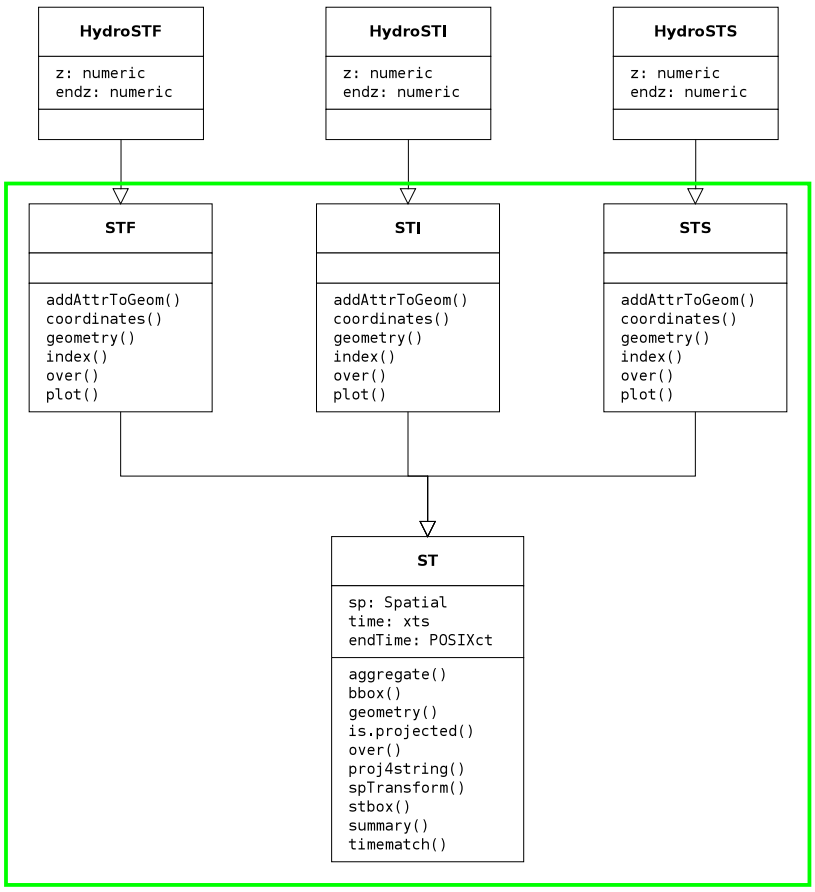


Figure 6.1.: Class diagram in the Unified Modeling Language (UML) showing the relationship between *spacetime* and *Hydro* classes. Classes defined in package *spacetime* are shown inside the green box.

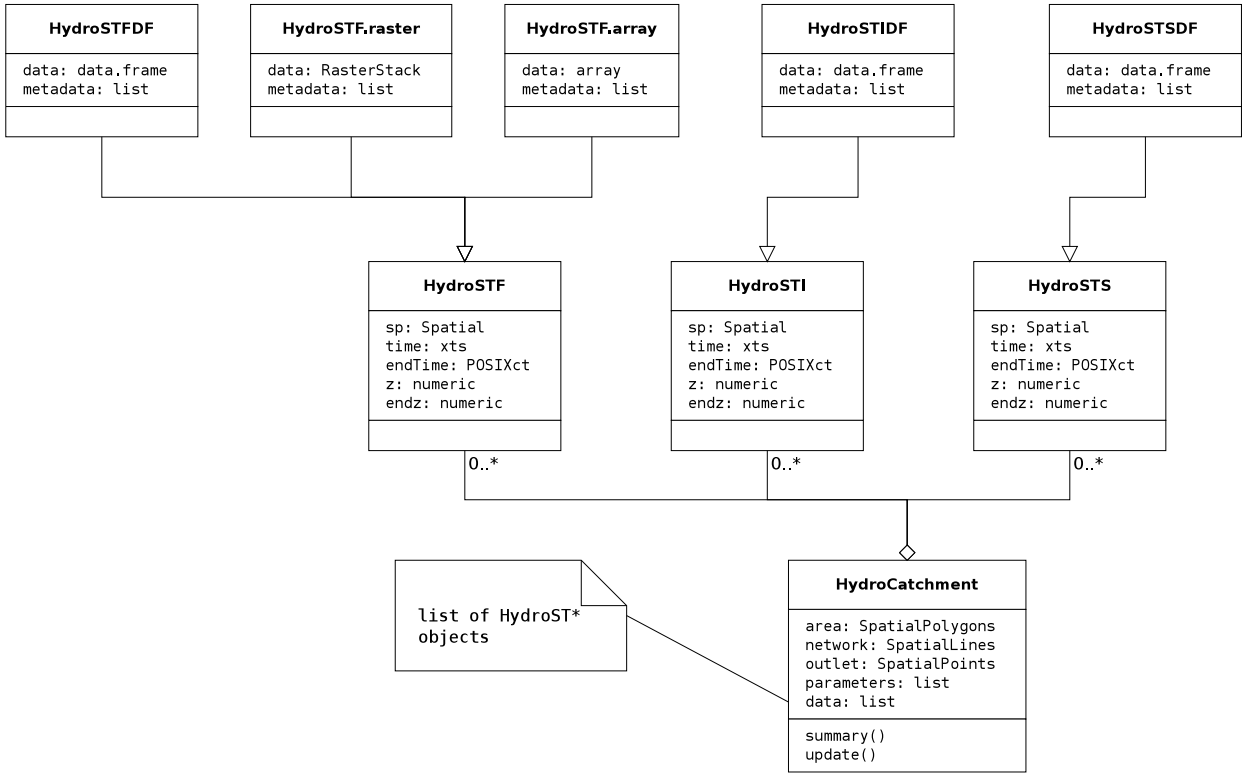


Figure 6.2.: Class diagram in the Unified Modeling Language (UML) for *Hydro*, showing the main classes and methods included in the package.

classes represent lattice layouts, in which spatial features have data values for multiple time points, class HydroSTI represents an irregular layout in which there is no defined space-time grid. In this case the number of data points equals n spatial features and m time points such that a data value k is associated with spatial feature k and time point k. Objects inheriting from HydroSTI therefore include a spatial feature and time point for each data value. The class can also be used when the structure of a data set is unknown and then promoted to HydroSTF if it meets the requirements of this class. Hydrological data with an irregular space-time layout include the time and location of burst pipes in a water supply network.

To avoid *ad hoc* solutions to model coupling there is a need for an intuitive way to group various model inputs so that they can efficiently be transferred between services within a workflow. A well-understood concept in hydrology is that of the catchment, which can be defined as the geographical area that contributes streamflow to certain point along a river system (Wagener et al., 2007). In *Hydro*, the HydroCatchment class represents data and information for a specific catchment in a standardised format. Slots `area`, `network` and `outlet` represent physical characteristics of a hydrological catchment. Data elements within HydroCatchment objects are objects deriving from HydroSTF, HydroSTS or HydroSTI classes and, therefore, include metadata entries from the CUAHSI controlled vocabulary. Slot `parameters` contains model parameters. Thus, service interfaces can be designed to receive a single HydroCatchment object, extract specific variables from the object using accessor (getter) functions, send it to the service in the required format, parse the output data from the service, update the HydroCatchment object using mutator (setter) functions and, finally, return the updated object to the workflow.

6.2.2 Methods

Methods for the *Hydro* data classes are shown in Table 6.1. Several methods are written for generic functions defined in other packages or base R for manipulating spatial, temporal or spatiotemporal data. The following examples use simple, artificial data objects to illustrate the key methods defined in the package.

Constructor functions HydroSTF, HydroSTI and HydroSTS create objects belonging to the respective classes. The class of the `data` argument, if provided, determines the derived class of the returned object. For example, if data is a RasterStack object, the constructor function will return an object of class HydroSTF.raster. Object metadata is provided as a named list where names correspond to the names of the CUAHSI controlled vocabulary tables. It is not necessary to provide object metadata but if metadata entries are supplied they are validated against the CUAHSI controlled vocabulary if the tables are available in

Table 6.1.: Methods included in the Hydro package

Function name	Description
crs	Coordinate reference system
diff	Time difference
dim	Dimensions
extent	Spatial extent
gridded	Test whether an object is spatially gridded
metadata	Get or set object metadata
names	Get or set object name
units	Get variable units

the current R session. Otherwise they are not validated and a warning is supplied. The controlled vocabulary tables can be downloaded from the CUAHSI WaterOneFlow Web service using functionality from package *RObsDat* (Reusser, 2016), as follows:

```
## Error in detach("raster", unload = TRUE): invalid 'name' argument
## Error in detach("spacetime", unload = TRUE): invalid 'name' argument
## Error in detach("sp", unload = TRUE): invalid 'name' argument
```

```
library(RObsDat)
library(RSQLite)
library(SSOAP)
file.remove("RODM.db")
con <- dbConnect(dbDriver("SQLite"), dbname = "RODM.db")
sqhandler <- new("odm1_1Ver", con=con)
options(odm.handler=sqhandler)
updateCV()

## Note:  method with signature 'PrimitiveSOAPType#list' chosen for function
'resolve',
##  target signature 'PrimitiveSOAPType#SchemaCollection'.
##  "SOAPType#SchemaCollection" would also be valid
```

Now the most recent version of the CUAHSI controlled vocabulary is available in the current R session. Package *RObsDat* includes various helper functions to navigate the controlled vocabulary tables. For example, the tables can be searched:

```
getMetadata(table="VariableName", Term="Discharge")

##        Term Definition
## 1 Discharge  Discharge
```

In addition, it allows users to add terms to controlled vocabulary tables:

```
addCV("VariableName", term="test", definition="test entry")
```

This facility should be treated with caution, however, because the term is only added to the local copy of the controlled vocabulary. Thus, when the table is updated from the WaterOneFlow Web service any locally added terms will be overwritten.

The following toy example creates objects `x` and `y` with classes `HydroSTFDF` and `HydroSTF.raster`, respectively. First, the space and time geometries are created together with some arbitrary data:

```
sp <- SpatialPoints(data.frame(x=1:5, y=1:5))
t <- seq(as.POSIXct("2000-01-01", tz="GMT"), by="1 hour", length.out=24)
dat <- data.frame(runif(n=120))
```

An object of class HydroSTFDF is created as follows:

```
x <- HydroSTF(STFDF(sp=sp, time=t, data=dat),
              metadata=list(VariableName="Discharge",
                            VariableUnitsID=52,
                            ValueType="Field Observation",
                            DataType="Cumulative"))
x

## class           : HydroSTFDF
## dimensions      : 5, 24, 1, 0 (space, time, depth, variables)
## spatial res.    : NA (points)
## temporal res.   : 1 hour
## spatial extent  : 1, 5, 1, 5 (xmin, xmax, ymin, ymax)
## temporal extent : 2000-01-01 GMT, 2000-01-01 23:00:00 GMT (start, end)
## coord. ref.     : NA
## data source     : in memory
##
##
```

```
## VariableName    :         Discharge
## VariableUnitsID :                52
## ValueType       : Field Observation
## DataType        :        Cumulative
##
##                            z1
## min values      : 0.01743481
## max values      :  0.9873524
```

To create an object of class HydroSTF.raster a RasterBrick object is created and passed to the constructor function, using the same time series as before:

```
dat <- brick(x=array(data=runif(2400), dim=c(10,10,24)))
y <- HydroSTF(data=dat, time=t,
              metadata=list(VariableName="Precipitation",
                            VariableUnitsID=52,
                            ValueType="Field Observation",
                            DataType="Cumulative"))
```

Note that in this case it is not necessary to supply a Spatial object to the constructor function because spatial reference information is derived internally from the RasterBrick object.

Coercion methods for objects belonging to HydroSTF, HydroSTS and HydroSTI classes make it easier to take advantage of methods defined for objects of different classes. Objects of class HydroSTF.raster and HydroSTF.array can always be coerced to HydroSTFDF objects, however, the reverse operation is only possible if the HydroSTFDF object is gridded. The following example coerces the HydroSTF.raster object created previously to an object of class HydroSTF.array:

```
y1 <- as(y, "HydroSTF.array")
class(y1)

## [1] "HydroSTF.array"
## attr(,"package")
## [1] "Hydro"
```

Next, the HydroSTF.array object is coerced to HydroSTF.raster and compare the result with the original HydroSTF.raster object:

```
y2 <- as(y1, "HydroSTF.raster")
compare(y, y2)

## [1] TRUE
```

The function `as.spacetime` converts objects inheriting from HydroSTF, HydroSTS, HydroSTI to their nearest equivalent *spacetime* classes. Then, the coercion methods of package *spacetime*, described in Pebesma (2012), can be applied. For example:

```
y1 <- as.spacetime(y)
class(y1)

## [1] "STFDF"
## attr(,"package")
## [1] "spacetime"

y2 <- as(y1, "xts")
class(y2)

## [1] "xts" "zoo"
```

Indexing allows users to extract or replace subsets of HydroST objects. Methods for HydroSTFDF, HydroSTSDF and HydroSTIDF objects closely follow those for the corresponding *spacetime* classes described by Pebesma (2012). Individual raster objects can be extracted from HydroSTF.raster objects using double brackets, as follows:

```
y[[1]]

## class       : RasterLayer
## dimensions  : 10, 10, 100  (nrow, ncol, ncell)
## resolution  : 0.1, 0.1  (x, y)
## extent      : 0, 1, 0, 1  (xmin, xmax, ymin, ymax)
## coord. ref. : NA
## data source : in memory
## names       : z1.1
## values      : 0.0008217387, 0.9898408  (min, max)
```

```
y[[1:3]]

## class       : RasterBrick
## dimensions  : 10, 10, 100, 3  (nrow, ncol, ncell, nlayers)
```

```
## resolution  : 0.1, 0.1  (x, y)
## extent      : 0, 1, 0, 1  (xmin, xmax, ymin, ymax)
## coord. ref. : NA
## data source : in memory
## names       :         z1.1,         z1.2,         z1.3
## min values  : 0.0008217387, 0.0066921259, 0.0005243178
## max values  :    0.9898408,    0.9974117,    0.9994207
```

The HydroCatchment class is designed to contain several data objects for a given place. To allow for the fact that providing information for area, network and outlet slots is not always necessary, a simple dummy object is included with the package. The following example loads this object into the current R session and updates it with the HydroSTFDF and HydroSTF.raster objects created previously:

```
data(hc)
hc <- update(hc, data=list(x, y))
names(hc)

## [1] "Discharge"     "Precipitation"
```

Index methods for HydroCatchment objects are designed to extract or replace objects held in the data slot. The index can be either be numeric or, if the objects have metadata entries, a character vector of variable names to be selected. Double brackets are used to select a single data object whereas single brackets can select one or more objects, returning a HydroCatchment object containing only the selected data:

```
x1 <- hc[[1]]
class(x1)

## [1] "HydroSTFDF"
## attr(,"package")
## [1] "Hydro"

y1 <- hc[["Discharge"]]
compare(x1, y1)

## [1] TRUE

class(hc[1])

## [1] "HydroCatchment"
## attr(,"package")
## [1] "Hydro"
```

```
class(hc[1, drop=TRUE])

## [1] "HydroSTFDF"
## attr(,"package")
## [1] "Hydro"
```

Spatial and temporal aggregation and overlay methods for HydroSTFDF, HydroSTIDF and HydroSTSDF objects are thin wrappers to the methods for objects of corresponding *spacetime* classes described in Pebesma (2012). Gridded HydroSTF objects can additionally be spatially aggregated to a lower resolution or resampled to a higher resoltion by taking advantage of the `aggregate` and `resample` methods of package raster. To achieve this, gridded HydroSTFDF and HydroSTF.array objects are coerced to HydroSTF.raster objects, aggregated, and coerced back again. Some examples of aggregation are shown here:

```
y1 <- aggregate(y, fact=2, FUN=mean)
dim(y1)

##     space      time     depth variables
##        25        24         1         0

y2 <- aggregate(y, fact=5, FUN=mean)
dim(y2)

##     space      time     depth variables
##         4        24         1         0

y2 <- aggregate(y2, by="2 hours", FUN=mean)
dim(y2)

##     space      time     depth variables
##         4        12         1         0
```

HydroST* objects can also be resampled to the spatial and temporal of another object. The next example creates a new raster object with the same spatial extent as the HydroSTF.raster object created earlier but a different spatial resolution. This object is used as the basis for resampling:

```
r <- raster(matrix(data=NA, nrow=5, ncol=5))
y1 <- resample(y, r, method="bilinear")
geometry(y1)
```

```
## class           : HydroSTF
## dimensions      : 25, 24, 1, 0 (space, time, depth, variables)
## spatial res.    : 0.2, 0.2
## temporal res.   : 1 hour
## spatial extent  : 0.1, 0.9, 0.1, 0.9 (xmin, xmax, ymin, ymax)
## temporal extent : 2000-01-01 GMT, 2000-01-01 23:00:00 GMT (start, end)
## coord. ref.     : NA
```

6.3 Example applications

6.3.1 Plynlimon

The purpose of this example is to demonstrate the usefulness of existing R packages, including *zoo*, *xts* and *spacetime*, as well as *Hydro*, for processing hydrological data. Observed temperature data from four automatic weather stations situated across the Plynlimon research catchments in Wales, United Kingdom, are used to calculate reference evapotranspiration using the Hargreaves equation, defined in the FAO Irrigation and drainage paper 56 (Allen et al., 1998) as follows:

$$ET_0 = 0.0023(T_{mean} + 17.8)(T_{max} - T_{min})^{0.5} Ra \tag{6.1}$$

where T_{mean} is mean temperature, T_{min} and T_{max} are minimum and maximum temperature, respectively, and Ra is extraterrestrial radiation. The original data files, obtained from the Centre for Ecology and Hydrology, were imported into R and saved as data frames in the R data package *HydroData*, which was developed for the purposes of storing data with which to experiment with *Hydro*. In the following code section the data frames of observations from the four weather stations, as well as various spatial data about the Plynlimon research catchments (coordinates of the four weather stations, catchment area, river network, gauging stations), are loaded into the workspace. The structure of one of the data frames is examined by printing its first six rows.

```
data(plyn_aws_carreg)
data(plyn_aws_eistg)
data(plyn_aws_gwy)
data(plyn_aws_tan)

data(plyn_area)
data(plyn_network)
data(plyn_flow_pts)
data(plyn_aws_pts)

head(plyn_aws_carreg)
```

```
##     SITE           HOUR_ENDED SOLAR_RADIATION NET_RADIATION WET_BULB_TEMP
## 1 carreg 03-JAN-1976 00:00:00               0            -3          2.11
## 2 carreg 03-JAN-1976 01:00:00               0            -7          1.48
## 3 carreg 03-JAN-1976 02:00:00               0           -10          0.97
## 4 carreg 03-JAN-1976 03:00:00               0            -5          0.82
## 5 carreg 03-JAN-1976 04:00:00               0            -8          0.37
## 6 carreg 03-JAN-1976 05:00:00               0            -6          0.17
##   DRY_BULB_TEMP WIND_SPEED WIND_DIRECTION RAINFALL
## 1          2.73     12.385            288      0.0
## 2          2.23     12.138            302      0.0
## 3          1.80     10.933            289      0.0
## 4          1.65     10.193            287      0.0
## 5          1.18     10.620            306       NA
## 6          0.92     10.001            305      0.5
```

The four data frames are combined into a list so that repetitive tasks can be performed in a loop, before printing the dimensions (number of rows, number of columns) of each data frame:

```
plyn_aws_list  <- list(plyn_aws_carreg,
                       plyn_aws_eistg,
                       plyn_aws_gwy,
                       plyn_aws_tan)
sapply(plyn_aws_list, dim)
```

```
##        [,1]   [,2]  [,3]   [,4]
## [1,] 306245 306461 99380 260336
## [2,]      9      9     9      9
```

Temperature is measured on an hourly basis but the records have different start and end points and, furthermore, have missing data points. Hence, the first processing task is to manipulate each data frame so that they refer to a common time series. The first command in the code section below creates a complete time series for the first four months of 2008. Each data frame in the list is converted to an *xts* time series object by coercing the first column, a character vector of time points, to a POSIXct object. The column containing temperature data is retained while columns with other variables are dropped. Each xts object is merged with the complete time series object, which effectively inserts missing value indicators at time points with missing data. The last command shows that the new objects have identical dimensions (one variable, 2,710 time points).

```
met.ts <- seq(from=as.POSIXct("2008-01-01 00:00:00", tz="Europe/London"),
              to=as.POSIXct("2008-04-22 22:00:00", tz="Europe/London"),
              by="1 hour")

for (i in 1:length(plyn_aws_list)) {
    x <- plyn_aws_list[[i]]
    x <- xts(x,
             order.by=as.POSIXlt(x$HOUR_ENDED,
                                 format="%d-%b-%Y %H:%M:%S"))
    x <- x["DRY_BULB_TEMP",drop=FALSE]
    x <- merge.xts(x, met.ts, join="right")
    plyn_aws_list[[i]] <- x
}

sapply(plyn_aws_list, dim)

##      [,1] [,2] [,3] [,4]
## [1,] 2710 2710 2710 2710
## [2,]    1    1    1    1
```

The xts objects in `plyn_aws_list` can now be combined into a single data frame. The data frames, which respectively store temperature data from four locations, are initially joined by rows (i.e. effectively stacked on top of one another) to create a spatio-temporal data frame in long format with time cycling fastest. The rows of this object are then reordered by the INDEX column, added specifically for this purpose, so that space cycles fastest in order to comply with the requirements of *spacetime* classes. These tasks are completed as follows:

```
for (i in 1:length(plyn_aws_list)) {
    x <- plyn_aws_list[[i]]
    x <- as.data.frame(x)
    row.names(x) <- NULL
    x <- cbind(x, data.frame(INDEX=seq(1,nrow(x))))
    plyn_aws_list[[i]] <- x
}

plyn_aws <- do.call(rbind, plyn_aws_list)
plyn_aws <- plyn_aws[order(plyn_aws$INDEX),]
```

The resulting data frame, **plyn_aws**, is used to create an STFDF object which is subsequently promoted to a HydroSTFDF object with relevant metadata.

```
stfdf <- STFDF(geometry(plyn_aws_pts), met.ts, plyn_aws)
T <- HydroSTF(stfdf[,,"DRY_BULB_TEMP",drop=FALSE],
              metadata=list(VariableName="Temperature",
                            VariableUnitsID=96,
                            ValueType="Field Observation",
                            DataType="Average"))
```

The Hargreaves method is based on estimates of minimum and maximum daily temperature. In the following code section the aggregate method for HydroSTFDF classes is used to estimate the minimum and maximum daily temperature across the four weather stations, as follows:

```
Tmin <- aggregate(T, by="1 day", FUN=min, na.rm=TRUE,
                  md=list(DataType="Minimum"))
Tmax <- aggregate(T, by="1 day", FUN=max, na.rm=TRUE,
                  md=list(DataType="Maximum"))
```

These two objects are grouped into a HydroCatchment object, using the spatial data loaded previously to specify catchment characteristics:

```
x <- HydroCatchment(area=plyn_area,
                    network=plyn_network,
                    outlet=plyn_flow_pts,
                    data=list(Tmin, Tmax))
```

One of the implications of associating HydroST data objects with metadata entries from a controlled vocabulary is to enable users to write functions that make "decisions" about

```
FAO56.eq52 <- function(x, ...) {

    Tmin <- getDataObject(x,
                          variablename="Temperature",
                          variableunitsid=96,
                          datatype="Minimum")
    Tmax <- getDataObject(x,
                          variablename="Temperature",
                          variableunitsid=96,
                          datatype="Maximum")

    if (!compare(Tmin, Tmax))
        stop("Arguments have different spatial/temporal characteristics")

    if (interval(Tmin, "days") != 1)
        stop("Arguments should have interval of at least 1 day")

    Ra <- FAO56.eq21(geometry(Tmin))

    Tmean <- (Tmin + Tmax) / 2

    ETref <- 0.0023 * (Tmean + 17.8) * ((Tmax - Tmin) ** 0.5) * Ra

    metadata(ETref) <- list(VariableName="Evapotranspiration",
                            VariableUnitsID=305,
                            ValueType="Model Simulation Result",
                            DataType="Unknown",
                            Model="FAO56 Equation 52 (Hargreaves)")

    update(x, data=list(ETref))
}
```

Figure 6.1: Function to calculate reference evapotranspiration using the Hargreaves equation. The input argument to the function is a HydroCatchment object containing minimum and maximum temperature.

```
FAO56.eq21 <- function(x, ...) {

    if (interval(x, "days") != 1) {
        stop("Argument x must have time step of 1 day")
    }

    jd <- as.numeric(format(as.Date(index(x@time),origin="1970-01-01"),'%j'))

    sp <- as(x@sp, "SpatialPoints")
    if (!raster::isLonLat(crs(x))) {
        sp <- sp::spTransform(sp, CRS("+proj=longlat +datum=WGS84"))
    }

    dr    <- 1 + 0.033 * cos((2 * pi / 365) * jd)

    delta <- 0.409 * sin((((2 * pi) / 365) * jd) - 1.39)

    lat   <- coordinates(sp)[,2,drop=TRUE] * pi / 180

    rad <- sapply(lat, FUN=function(lat1) {
        ws <- acos(-tan(lat1) * tan(delta))
        rad <- 24 * 60 / pi * 0.0820 * dr *
            (ws * sin(lat1) * sin(delta) + cos(lat1) * cos(delta) * sin(ws))
        rad})

    rad <- data.frame(Radiation=as.numeric(t(rad)))

    HydroSTF(STFDF(sp=x@sp, time=x@time, data=rad, endTime=x@endTime),
             metadata=list(VariableName="Global Radiation",
                           VariableUnitsID=144,
                           ValueType="Estimated Value",
                           DataType="Cumulative"))
}
```

Figure 6.2: Function to calculate extraterrestrial radiation using FAO56 equation 21. The input argument to the function is an object inheriting from STF.

the necessary steps to perform a set of tasks given the information provided by the metadata. A simple example of this might be to automatically check and, if necessary, convert the units of input variables to an application in order to ensure they are compatible. To further demonstrate this capability the Hargreaves equation was implemented as a function, `FAO.eq52`, which accepts a single HydroCatchment object. The function is shown in Figure 6.1. As well as estimates of minimum and maximum temperature the Hargreaves method requires extraterrestrial radiation. This is calculated from the latitude and day of year, which can be derived from the spatio-temporal geometry of the HydroSTFDF objects representing temperature. As Figure 6.1 shows, the process followed by `FAO.eq52` is to check for the existence of solar radiation in the HydroCatchment object and, if it is not present, extract the spatio-temporal geometry of one of the HydroSTF objects to an implementation of FAO56 Equation 21, which calculates solar radiation from latitude. This function uses several methods defined in *Hydro* to extract the latitude and Julian day of each data point in the HydroSTF object. Having calculated solar radiation it is straightforward to calculate reference evapotranspiration. The original HydroCatchment object, updated with the object containing reference evapotranspiration, is returned.

```
x <- FAO56.eq52(x)
VariableName(x)

## [1] "Temperature"        "Temperature"        "Evapotranspiration"
```

Lastly, a simple visualisation of reference evapotranspiration over the Plynlimon catchments is achieved by aggregating values across the four weather stations:

```
Hargreaves <- getDataObject(x,
                            variablename="Evapotranspiration",
                            Model="FAO56 Equation 52 (Hargreaves)")
Hargreaves <- aggregate(Hargreaves, by="space", FUN=mean, na.rm=TRUE)
xyplot(Hargreaves)
```

The resulting plot, with some refinements, is shown in Figure 6.3.

6.3.2 Reservoir operations

Reservoir operations for water supply follow either a standard operating policy, whereby demand is met unless it exceeds water supply, or a hedging policy, under which the release of water is reduced before a deficit occurs to safeguard against severe water shortage (You and Cai, 2008). Recently, Garcia et al. (2016) proposed a stylised socio-hydrological model

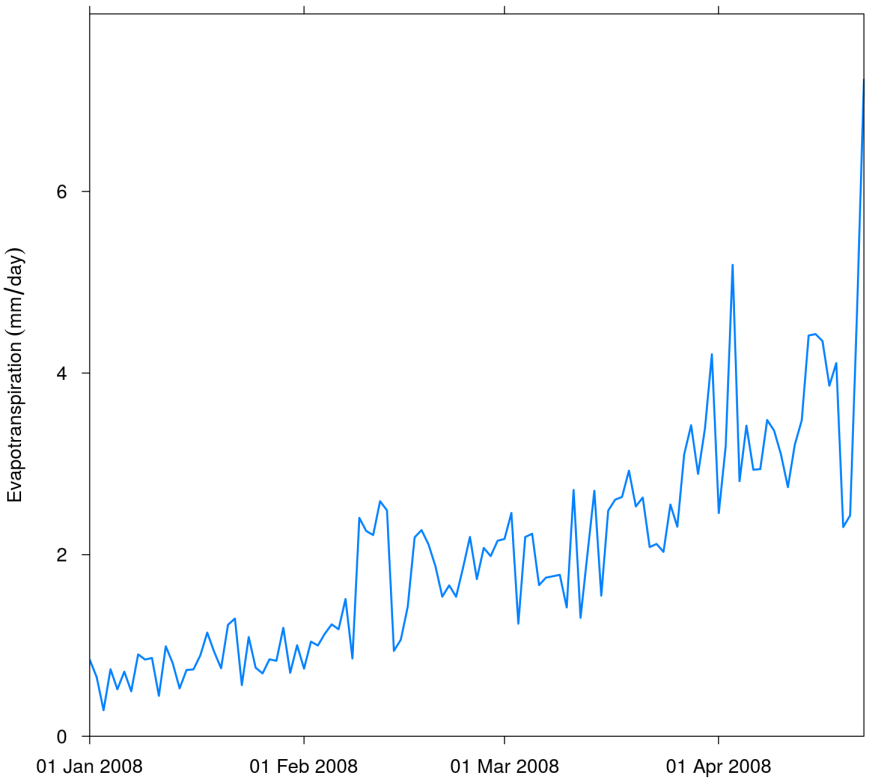


Figure 6.3.: Mean reference evapotranspiration during the first four months of 2008 across four weather stations in the Plynlimon catchments in Wales, United Kingdom. Reference evapotranspiration was estimated using the Hargreaves method.

focusing on the question "what is the impact of reservoir operation policy on the reliability of water supply for a growing city?". The model, which is fully described in Garcia et al. (2016), includes a proposed feedback mechanism between the reliability of water supply and population growth. Here, the classes defined in *Hydro* are used to develop a component-based implementation of the reservoir operations model in order to explore the potential utility of the R system for model coupling. Equations 6.2–6.8 are reproduced from Garcia et al. (2016) for the sake of completion: readers are advised to consult the original source for a full explanation of the model formulation.

The main input to the reservoir operations model is the annual streamflow into the reservoir, Q. In Garcia et al. (2016) this is modelled using a first-order regressive model, as follows

$$Q_t = \rho_H \left(Q_{t-1} - \mu_H\right) + \sigma_H \left(1 - \rho_H^2\right)^{0.5} a_t + \mu_H \tag{6.2}$$

where t is time, μ_H is mean discharge, σ_H is standard deviation, ρ_H is autocorrelation with lag one and a_t is a random variable drawn from a normal distribution ($\mu = 0, \sigma = 1$). At every time step the the change in the volume of water held in the reservoir, V, is given by the water balance equation

$$\frac{dV}{dt} = Q_t - W_t - \eta_H A_t - Q_D - Q_E \tag{6.3}$$

where W is the amount of water withdrawn from the reservoir, η_H is evaporation, A is the reservoir area, Q_D is downstream allocation and Q_E is environmental flow . Water demand in the model is driven by population, P, which changes according to average rates of birth (δ_B), death (δ_D), immigration (δ_I) and emigration (δ_E). Immigration and emigration are influenced by the perception of water shortage, M, following Sterman (2000). A threshold, τ_P, is introduced to account for the fact that perception of water storage only influences population at high levels, so that

$$\frac{dP}{dt} = \begin{cases} P_t[(\delta_B - \delta_d) + \delta_I(1 - M_t) - \delta_E(M_t)], & M_t \geq \tau_P. \\ P_t[\delta_B - \delta_D + \delta_I - \delta_E], otherwise. \end{cases} \tag{6.4}$$

The amount of water withdrawn from the reservoir depends on the operation policy in use. Withdrawal decisions for time t are made based on the predicted streamflow, $(1-f) \times Q_t$, where f is a factor ($0 < f < 1$), representing the proportion of streamflow diverted to meet the downstream allocation and environmental flow requirements. Under the hedging policy withdrawals are gradually reduced once the volume of the reservoir falls below a critical threshold, $K_P D_t P_t$, where K_P is the hedging slope and D is water use per capita. Conversely, under the standard operating policy there is no hedging ($K_P = 1$) so the withdrawal equals the water use per capita multiplied by population, $D_t P_t$, unless this exceeds the total amount of water in the reservoir. If, after accounting for total water demand, downstream allocation and environment flow requirements, the stored water exceeds the maximum capacity, V_{max}, excess water is spilled, effectively contributing to water withdrawal. This information can be expressed as

$$W_t = \begin{cases} V_t + (1-f)Q_{t-1} - V_{max}, & V_t + (1-f)Q_{t-1} \geq D_t P_t + V_{max} \\ D_t P_t, & D_t P_t + V_{max} > V_t + (1-f)Q_{t-1} \geq K_P D_d P_t \\ \dfrac{V_t + (1-f)Q_{t-1}}{K_P}, & K_P D_t P_t > V_t + (1-f)Q_{t-1}. \end{cases} \tag{6.5}$$

where V_{max} is the maximum volume of the reservoir and all other variables are previously defined. If W_t is less than the total demand, $D_t P_t$, there is a water shortage, S_t, such that

Table 6.2.: Names of model components of reservoir operations model of Garcia et al. (2016) with reference to the corresponding model equation. In this example each component is represented as an R function, although there is no requirement for this to be the case.

Component	Equation
stage_storage_component	-
reservoir_surface_area_component	-
reservoir_allocation_component	-
downstream_allocation_component	-
environmental_flow_component	-
evaporation_component	-
streamflow_component	6.2
volume_component	6.3
population_component	6.4
water_withdrawal_component	6.5
water_shortage_component	6.6
memory_component	6.7
demand_component	6.8

$$S_t = \begin{cases} D_t P_t - W_t, & D_t P_t > W_t \\ 0, & otherwise. \end{cases} \tag{6.6}$$

A water shortage event leads to a change in the level of perception of water shortage, M, as follows

$$\frac{dM}{dt} = \left(\frac{S_t}{D_t P_t}\right)^2 (1 - M_t) - \mu_S M_t \tag{6.7}$$

where μ_S is a parameter which the controls the rate at which the level of awareness of water shortage decreases over time. An awareness of water shortage leads to a change in the per capita demand, according to the equation

$$\frac{dD}{dt} = -D_t \left[M_t \alpha \left(1 - \frac{D_{min}}{D_t}\right) + \beta \right] \tag{6.8}$$

where α is the rate of adoption of more efficient practices resulting from a water shortage event, β is the background rate at which water use efficiency improves and D_{min} is the minimum per capita water demand.

To demonstrate the use of *Hydro* for orchestrating scientific workflows involving coupled human-water systems the reservoir operations model was implemented as a set of model components representing processes described in the above equations. The new implementation was adapted from the original R script developed by Garcia et al. (2016), which is avail-

```
waterShortageComponentWrapper <- function(x, t0, ...) {
    w <- getDataObject(x, variablename="Water Use, Public Supply")
    d <- getDataObject(x, variablename="Water demand")
    p <- getDataObject(x, variablename="Population")
    s <- getDataObject(x, variablename="Water shortage")
    s[,t0] <- water_shortage_component(w[,t0,drop=TRUE],
                                       d[,t0,drop=TRUE],
                                       p[,t0,drop=TRUE])
    update(x, data=list(s))
}

water_shortage_component <- function(withdrawal, demand, population, ...) {
    test <- demand * population > withdrawal
    yes <- demand * population - withdrawal
    no <- rep(0, length(withdrawal))
    result  <- yes
    result[!test] <- no[!test]
    result
}
```

Figure 6.3: Example model component and wrapper function. For simplicity, the model component is an R function but in reality it could be implemented in any language and have a wide range of interfaces.

able as a Web application (`https://mgarcia.shinyapps.io/ReservoirOperations/`). Table 6.2 shows the names of the model components and the equations they implement. In the interests of clarity, each component is written as a plain R function with standard numeric input arguments. A wrapper function for each component was provided as an interface between the model component and the HydroCatchment object. For example, Figure 6.3 shows the component and wrapper to calculate water shortage according to Equation 6.6. The first stage in the workflow is to create HydroSTFDF objects for model variables and to group these in a HydroCatchment object. The convention employed currently is to supply preallocated HydroSTF objects representing variables that are modified during the model run. Initial values are supplied for the first time point in the HydroST* objects where applicable. For example, the HydroSTFDF object representing discharge is created with the commands:

```
t0 <- as.POSIXct("2010-01-01 00:00:00", tz="GMT")
ts <- seq.POSIXt(t0, by="1 year", length.out=56)
sp <- SpatialPoints(data.frame(x=1, y=1))
Q <- HydroSTF(xts(rep(NA, length(ts)), order.by=ts), sp=sp,
```

```
                metadata=list(VariableName="Discharge",
                              VariableUnitsID=126,
                              ValueType="Model Simulation Result",
                              DataType="Cumulative"))
```

This step was repeated for eleven additional variables used in the coupled model. Since the reservoir operations model is not spatially explicit, an arbitrary spatial point is used to comply with the HydroST* class specification. The reservoir operations model requires initial values for discharge into the reservoir, Q, reservoir storage, S, population served by the reservoir, P, awareness of water shortage amongst the population, M, and per capita water demand, D. The respective HydroSTFDF objects are updated in the following way:

```
S[,1:2] <- 0.5
P[,1:2] <- 1e+06
M[,1:2] <- 0
D[,1:2] <- 4e-07
Q[,1:2] <- 2
```

Parameters required for the reservoir operations model are described in Garcia et al. (2016, Table 5). HydroCatchment objects currently represent parameters as a named list:

```
params <- list(rhoH=0.6,       ## streamflow lag one autocorrelation
               muH=2.0,        ## mean streamflow
               Efrac=0.25,     ## environmental flow allocation
               Dfrac=0.5,      ## downstream allocation
               Rfrac=0.25,     ## reservoir allocation
               sigmaH=0.5,     ## standard deviation streamflow
               baseArea=10,    ## reservoir base area
               minElev=-0.05, ## minimum elevation
               maxElev=0.5,    ## maximum elevation
               Tp=0.4,         ## threshold
               dt=1,
               sigmaT=0.1,     ## average slope of reservoir
               Kp=1,           ## hedging policy
               etaH=0.001,     ## evaporation rate
               muS=0.05,       ## awareness loss rate
               alpha=0.15,     ## fractional efficiency adoption rate
               beta=0.001,     ## background efficiency rate
               Vmax=2.0,       ## maximum reservoir capacity
               Dmin=2e-07,     ## minimum water demand
               deltaB=0.04,    ## birth rate
               deltaD=0.03,    ## death rate
               deltaI=0.05,    ## immigration rate
               deltaE=0.03)    ## emmigration rate
```

Thus, having preallocated the model variables and defined parameter values, it is possible to create the HydroCatchment object representing the reservoir operations model domain. In the following code snippet a template HydroCatchment object containing arbitrary values for slots `area`, `network` and `outlet` is updated with the data required to run the model:

```
data(hc)
hc <- update(hc, parameters=params,
             data=list(S, P, M, D, Q, Qr, Qd, Qe,
                       ResArea, Withdrawal, Shortage, Evaporation))
```

As shown in Figure 6.4, the reservoir operations model is expressed as an R function receiving a HydroCatchment object and a time series which defines the temporal grid of the simulation. Thus, the coupled model is run with the following command:

```
hc <- reservoirOperationsModel(hc, ts)
```

returning an updated version of the HydroCatchment object created above. The model results are summarised in Figure 6.4, where the impacts of the feedback mechanism proposed by Garcia et al. (2016) are clearly apparent. For example, it can be seen that around 2040, when reservoir storage approaches zero, the awareness of water shortage increases and, consequently, the per capita demand also decreases. After a decade where reservoir storage fails to recover its initial value, population stabilises and, after a further decade, reservoir storage begins to increase.

```
reservoirOperationsModel <- function(x, time, ...) {

    x <- stageStorageCurveComponent(x)
    x <- reservoirAllocationComponentWrapper(x, t0=time[1])

    for (i in 2:(length(time) - 1)) {

        t0 <- time[i]
        t1 <- time[i+1]

        ## calculate fluxes
        x <- reservoirSurfaceAreaWrapper(x, t0=t0)
        x <- reservoirAllocationComponentWrapper(x, t0=t0)
        x <- downstreamAllocationComponentWrapper(x, t0=t0)
        x <- environmentalFlowComponentWrapper(x, t0=t0)
        x <- waterWithdrawalComponentWrapper(x, t0=t0)
        x <- waterShortageComponentWrapper(x, t0=t0)
        x <- evaporationComponentWrapper(x, t0=t0)

        ## feedbacks
        x <- volumeComponentWrapper(x, t0=t0, t1=t1)
        x <- populationComponentWrapper(x, t0=t0, t1=t1)
        x <- memoryComponentWrapper(x, t0=t0, t1=t1)
        x <- demandComponentWrapper(x, t0=t0, t1=t1)

        ## simulate streamflow for the subsequent time point
        x <- streamflowComponentWrapper(x, t0=t0, t1=t1)
    }
    x
}
```

Figure 6.4: Function to orchestrate components of the reservoir operations model.

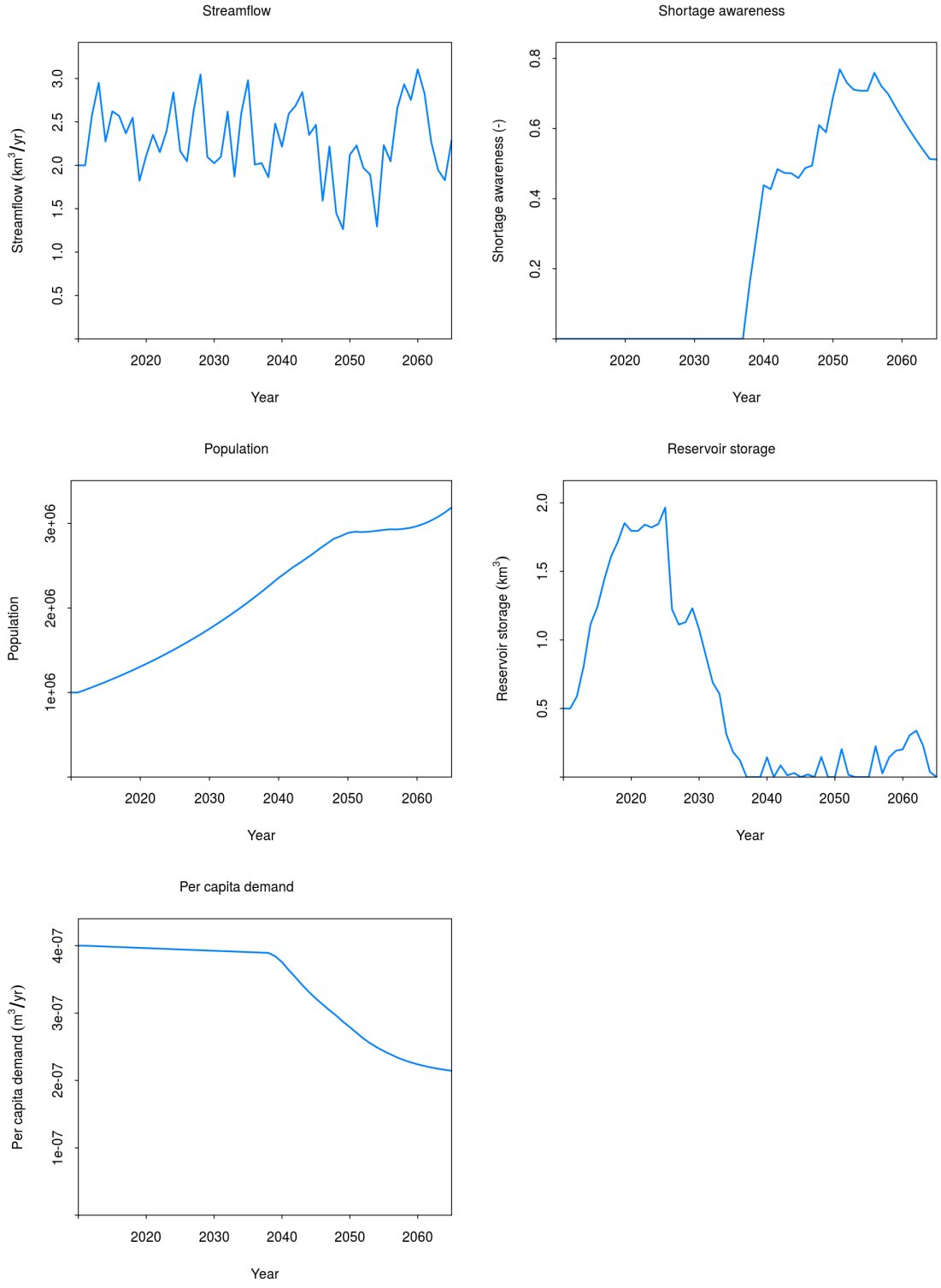


Figure 6.4.: Results of the reservoir operations model simulation, showing the feedback mechanisms proposed by Garcia et al. (2016). Around 2040, when reservoir storage approaches zero, the awareness of water shortage increases and, consequently, the per capita demand also decreases. After a decade where reservoir storage fails to recover its initial value, population stabilises and, after a further decade, reservoir storage begins to increase.

6.4 Discussion

The *Hydro* software package implements a set of classes and methods designed to facilitate hydrological data analysis and the development and orchestration of scientific workflows. The package enables the R system to be used as an environmental modelling framework providing, in theory at least, support for 'component interaction and communication, spatial-temporal stepping and iteration, up/downscaling of spatial data, multithreading/multiprocessor support, and cross language interoperability, as well as reusable tools for data analysis and visualization' (Lloyd et al., 2011). Modelling frameworks should enable the integration of models and data transparently and in a way that is reproducible not only by the original author but also by scientists and practitioners from diverse research communities. In practice, this is achieved by enabling semantic, methodological and technical integration between models (Knapen et al., 2013). Semantic integration results in a common understanding between models about the facts and concepts represented by shared data resources (Knapen et al., 2013). In *Hydro* this is achieved in various ways. Most obviously, the use of a controlled vocabulary to describe data provides an unambiguous definition of the variable represented by HydroST* objects. Since the metadata entries are predetermined it is possible for developers to write applications which act upon the information contained in the metadata entries, as demonstrated in the Plynlimon example. The package currently relies on the CUAHSI controlled vocabulary. However, the reservoir operations example demonstrates the limitations of this approach because, as a system originating from the hydrological sciences, it does not support many of the terms that might be encountered in other disciplines. It is possible to suggest new entries to the vocabulary but it is unclear whether there is an ambition to extend its scope to include terms from other domains. In any case, given the range of problems encountered in hydrology, particularly in socio-hydrology, it is unlikely that a single vocabulary will contain all the relevant terms all of the time. Thus, future work should provide the technical infrastructure to incorporate controlled vocabularies from multiple domains. Semantic integration is also achieved through the use of standardised data formats to represent space and time attributes in HydroST* objects. Time is stored as an object of class `xts` in which the underlying temporal index conforms to POSIX standards (Ripley and Hornik, 2001), while space is represented by objects of the classes defined in package *sp*, which interfaces to GDAL, OGR and PROJ.4 libraries. This ensures the spatial and temporal characteristics of variables can be communicated to models in an unambiguous way and acted upon by a range of applications.

Methodological integration allows the exchange of information between models which may operate at different temporal and spatial scales (Liu et al., 2008; Knapen et al., 2013). It is achieved in *Hydro* through the definition of methods for the underlying data classes,

HydroSTF, HydroSTS and HydroSTI, enabling spatial, temporal and spatiotemporal aggregation and disaggregation, spatial reprojection as well as geostatistical interpolation and stochastic simulation (Pebesma, 2004). Technical integration is the automated transfer of data between model components in order to create fully coupled model (Argent, 2004; Knapen et al., 2013) which, in *Hydro*, is provided through the scripting capabilities of the R system and demonstrated by the reservoir operations model described above and shown in Figure 6.4. While the example application is relatively simple, not least because it is spatially lumped, it shows the flexibility of scripting languages for the orchestration of scientific workflows. In this way the approach followed by Hydro follows that of the OMS3 system, which requires modellers to write applications in the Groovy language, rather than the GUI approach of the OpenMI Configuration Editor. The main advantage of this approach compared to GUI-based solutions is its flexibility, which affords a great deal of control to application developers. In doing so, however, it requires a certain level of programming knowledge and may present a steep learning curve to some potential users. It also means that many tasks which may be implicitly performed by an orchestration engine such as OpenMI Configuration Editor, such as regridding, may have to be defined explicitly. One of the advantages of using the R system, however, is that it is used by an active community of geoscientists for a wide variety of computational tasks (Pebesma, 2012).

The reservoir operations model combines several model components to provide a representation of a hypothetical system in which water shortage leads to a decline in the per capita water consumption and the population growth rate (Garcia et al., 2016). Each model component consists of the original model code, in this case simply expressed as an R function receiving data types defined in base R (i.e. numeric, POSIXct), and a wrapper function which provides an interface between HydroCatchment objects and the model. The wrapper extracts from the HydroCatchment object the information required by the model component, formats the data accordingly, parses the model output and updates and returns the HydroCatchment object. There is no requirement for models to be written in the R language. It should be possible, at least in principle, to support models written in other programming languages and conforming to the standards of other modelling frameworks. The R system has convenient APIs for software written in various languages including C/C++, Fortran and Java. Thus, writing an interface between the HydroCatchment class and external code should be straightforward in most cases. Equally, executable files can be run from within R by invoking system commands through the `system` interface. Several hydrological models are already available in the R system. For example, package *topmodel* (Buytaert, 2011) provides a convenient wrapper to the original Fortran code of Beven and Kirkby (1979), while package *SWATmodel* provides an implementation of the Soil and Water Assessment Tool. The FUSE toolbox (Clark et al., 2008), which includes imple-

mentations of Precipitation Runoff Modeling System (Leavesley et al., 1983), Sacramento model, Topmodel and Variable Infiltration Capacity (ARNO/VIC) model, is also available.

Cloud computing is the use of computer hardware and software delivered as a service over the Internet to store, manage and process data (Bürger et al., 2012). This enables the efficient utilisation of computer infrastructure by multiple users and removes the need to install and maintain various software components locally (Bürger et al., 2012). The use of distributed and cloud computing in scientific workflows has generated significant interest in recent years (e.g. Jha et al., 2011; Goodall et al., 2011; Peckham and Goodall, 2013). As Buytaert et al. (2012) point out, 'the complexity of most environmental models typically confines them to scientific laboratories and academic computer clusters'. This is especially the case for distributed, process-based models which typically require expert knowledge just to install the model code. Cloud computing will be important to ensure community engagement and participation with the development of so-called "hyper-resolution" models because the computing power required for such models will exceed that provided by desktop computers. As the principles of socio-hydrology are directed towards global and regional problems, envisaged by (Lall, 2014), model complexity will increase. There remain considerable difficulties with developing model encapsulations for Web services, as discussed by Yue et al. (2015). However, the principle for writing a wrapper for a model exposed as a Web service would the same as that for locally installed code. It should extract the necessary data from the HydroCatchment object, format and send information according to the specific protocol of the Web service, receive Web service output and update the HydroCatchment object. The R system includes low-level packages for accessing Web services including RCurl and SSOAP as well as high-level packages providing access to specific Web services. One such package is *WaterML*, which allows users to retrieve data from the CUAHSI Hydrologic Information System (Kadlec et al., 2015).

The concept of the Model Web envisages 'a dynamic web of models, integrated with databases and websites, to form a consultative infrastructure where researchers, managers and policy makers and the general public can go to gain insight into "what if" questions' (Bastin et al., 2013; Geller and Turner, 2007). It would enable the integration of resources, including data and models, to build coupled models of complex systems where components are loosely coupled over a distributed system of networked computers (Geller and Melton, 2008; Goodall et al., 2011; Bastin et al., 2013). The Model Web is based on a service-oriented architecture in which models and databases are exposed as interoperable Web services (Castronova et al., 2013; Nativi et al., 2013; Schade et al., 2012; Zou et al., 2012). Individual services within a network are regarded as black-boxes and may have no awareness of other services (Barbosa and Barbosa, 2009). The example application of *Hydro* to the reservoir operations model can be seen as an implementation of the Model Web

without web services. The realisation of the Model Web is highly ambitious and will rely on bottom-up, community driven progress. According to Nativi et al. (2013), the Model Web is based on four main principles: 'open access, minimal barriers to entry, service driven and scalability'. Thus, any solution meeting these criteria is 'a step towards the long term vision' (Nativi et al., 2013). Therefore, future work on *Hydro* will develop use-cases with interacting Web services. To expose models as Web services the 52 North Web Processing Service (WPS; Hinz et al., 2013), based on the Open Geospatial Consortium WPS interface standard and providing processing backends for R, Java, Python, GRASS GIS and ArcGIS, shows great promise.

There are several aspects of *Hydro* which require further development. Considering HydroST* classes, there needs to be a way to represent the uncertainty associated with variables and, similarly, a way of representing the same variable generated by different models without having to create a separate object. In practice, this would mean adding a fifth dimension to HydroST* classes to represent different realisations of the same variable. Thus, there would be a need for a standard way of referencing individual realisations, raising the question of whether there is a need for a controlled vocabulary to refer to specific models. A further limitation of *Hydro* in its current form is that it is not well adapted to handling large datasets because R requires all data to be stored in memory. The use of SciDB, an array database system for which the plugin *scidb4geo* provides spatial and temporal reference information, shows promise for managing large spatio-temporal datasets. Effectively in this case the data slot of the HydroSTF class would be a database connection and methods would be written to leverage the existing interface between R and SciDB instances provided by package *scidb*.

Workflow orchestration using *Hydro* works by providing wrappers to the original model. The main advantage of this approach over modelling frameworks such as OMS and OpenMI is that it does not require any modifications to the original model code. The main area for improvement for the current software implementation is to reduce the computational effort required to execute workflows. To run the original model of Garcia et al. (2016), which may be viewed as a tightly coupled application, on a 64-bit machine with Intel Core i3 @ 1.4 GHz and 4Gb RAM takes approximately 5 seconds, whereas the loosely coupled version presented in the example application above takes around 20 seconds. While in this example the additional time is merely an inconvenience, it is not difficult to conceive of applications that would quickly become prohibitively time consuming using the approach outlined above. Computational inefficiency mainly arises from the overhead associated with continually stopping and starting the model, such as reading and writing model input/output files, as well as the need to get and set HydroCatchment data at each time point. There are also some improvements to be made to the design of the Hydro classes.

In particular, the treatment of parameters in HydroCatchment objects seems cumbersome because it effectively relies on the list of parameters following a naming convention. An alternative approach would be to design a separate class, which could be called HydroParameters, from sub-classes representing specific models could be derived. This would have the advantage that parameter values could be checked before they were supplied to models.

7 Summary and conclusions

7.1 Summary of thesis

7.1.1 Modelling land use/land cover change

Spatially explicit data about historical land use/land cover change are an essential input to assessments of environmental change. Maps showing historical land use/land cover additionally provide important insight into the drivers of change at regional and local scales, which can subsequently inform planning decisions at various administrative levels and constrain predictions about the quantity and location of future change. While contemporary land use/land cover maps are commonly obtained by classifying remotely sensed data, the quality and availability of satellite images over much of the Earth's surface before the year 2000 is poor. As a result, existing datasets of historical land use/land cover at global and regional scales have used various spatial disaggregation algorithms to combine contemporary land use/land cover maps with national and sub-national inventory data (e.g. Klein Goldewijk et al., 2011; Tian et al., 2014). These approaches are fundamentally based on the assumption that the relative spatial distribution of land use/land cover remains constant in time. However, this approach fails to account for the spatial and temporal variability of the underlying drivers of change. Dynamic land use change models, which simulate the rate and location of change using various approaches, have been suggested as a way to improve historical land use/land cover mapping (Tian et al., 2014).

A detailed literature review at the beginning of Chapter 2 revealed several key limitations of existing software tools for land use change modelling. In particular, it was found that currently available software applications usually only perform the spatial allocation of change, necessitating the use of additional software to process input data, carry out statistical analysis and validate model outputs. Section 2.2 implemented a new modelling framework, *lulcc*, which addresses many of the identified limitations. The new software:

- Provides a framework which allows every part of the modelling workflow to be carried out within the same environment;
- Implements multiple allocation routines within a common framework, facilitating

model intercomparison and ensemble experiments;

- Includes a state of the art validation method which makes it easier for modellers to calibrate and validate their models and communicate the results with sophisticated methods for visualisation.

Utilising the new modelling framework, Chapter 3 developed a regional-scale application of the Change in Land Use and its Effects (CLUE) spatial allocation routine (Veldkamp and Fresco, 1996a; Verburg et al., 1999) to reconstruct historical land use/land cover in India between 1956–2010. Random forest predictive models were used to describe the quantitative relationship between the spatial distribution of contemporary land use/land cover, derived from a state of the art contemporary product, and selected biophysical and socioeconomic covariates. Available satellite data were assimilated into the analysis by including metrics derived from the NDVI3g dataset (Pinzon and Tucker, 2014) as covariates in the statistical models of vegetated land use/land cover categories. The resulting dataset has a 5 arcminute $\times$ 5 arcminute spatial resolution and shows the fractional membership of each grid cell in the study region to forest, urban, agricultural uses, grassland, barren/sparsely vegetated land, water and permananent snow and ice on an annual basis during the study period. Post-processing was used to separate agricultural uses into cropland, fallow land and tree crops and grassland into grassland, shrubland and wetland, in order to facilitate the transformation of the dataset into maps of land cover.

7.1.2 Mapping irrigated land

Agricultural intensification across South Asia has resulted in substantial land change from low intensity rainfed cropland to high-input agricultural systems. In India, for example, while the area devoted to agricultural land uses has grown by around 5% between 1956–2010, net irrigated area has grown by more than 60% over the same period, with much of this change happening across the Indo-Gangetic plains in northern India. As a result, spatially explicit data about the growth in irrigated area over time is necessary to quantify the impact of irrigation activities on regional water resources and climate. Chapter 4 spatially disaggregated district-level agricultural inventory data about the irrigated area of various crop types in India to a 5 arcminute $\times$ 5 arcminute spatial grid using maps of historical cropland extent, developed in Chapter 3, and biophysical suitability maps for various irrigated crops obtained from the FAO Global Agro-Ecological Zones database (IIASA, 2012). The advantages of the resulting dataset compared to previous attempts to map irrigated land are as follows:

- It considers the biophysical suitability of grid cells to individual crops, providing a physical basis for the spatial allocation of district-level agricultural inventory data;

- In contrast to several previous irrigated area datasets which show the area equipped for irrigation it shows the area actually irrigated; a quantity which is more closely related to water consumption;
- It provides the irrigated area of 25 crop types, allowing modellers to take into account the physical characteristics and growing season of individual crops in assessments of environmental change.

Validation of the dataset at the grid square level for historical periods was impossible because there are no alternative datasets showing the change in irrigated area of specific crops over time. However, the maps for the year 2000 show good agreement the corresponding maps from the MIRCA2000 dataset (Portmann et al., 2010). Furthermore, they are consistent with district-level agricultural inventory data.

7.1.3 Land surface modelling of the Ganga basin

Land change influences the surface energy and water budget in various ways (Feddema, 2005). In India, several observational and modelling studies have pointed towards the existence of a feedback mechanism between elevated soil moisture values across the Indian subcontinent as a result of irrigation and the behaviour of the South Asian monsoon. Historical assessments using models are necessary to gain further insight into land-atmosphere feedbacks and improve predictions about the future behaviour of the monsoon. Previous modelling studies to quantify the effect of irrigation on the South Asian monsoon are associated with considerable uncertainty because of limitations of the available data about historical land use/land cover change and irrigated area. Chapter 5, therefore, integrated the land use/land cover change dataset developed in Chapter 3 with the Joint UK Land Environment Simulator (JULES; Best et al., 2011; Clark et al., 2011b), a process-based land surface model, in order to simulate historical soil moisture values in the Ganga basin for the period 1971–2005. However, since the version of JULES available at the time of the experiment did not represent irrigation processes, it was necessary to devise an alternative methodology for simulating the effects of irrigation on the land surface. A detailed literature review showed that existing approaches typically define a threshold value below which the soil moisture in irrigated areas is not allowed to fall (e.g. Lobell et al., 2009; Saeed et al., 2009; Tuinenburg et al., 2014). A key limitation of existing methods is the fact that irrigation is commonly assumed to occur for the entire year without consideration of the growing season of individual crops. Moreover, irrigated areas are usually defined according to the Global Map of Irrigated Areas dataset (Siebert et al., 2005) which shows the area equipped for irrigation rather than the area actually irrigated.

Section 5.2 developed an improved methodology which exploited the novel characteristics of the irrigated area dataset developed in Chapter 4. Crop calendars obtained from MIRCA2000 and Aquastat were used to temporally disaggregate the irrigated area dataset from annual maps to monthly maps based on the growing season of specific crops. Simulated soil moisture values from JULES were then modified at each time point based on the assumption that soil moisture in irrigated areas is minimally kept at field capacity. This resulted in two soil moisture datasets: one considering land use/land cover change only and one which additionally takes into account the effects of irrigation. Within the context of Hydroflux India these datasets were used to force a regional climate model in order to isolate the effects of large-scale irrigation across the Gangetic plain on the behaviour of the South Asian monsoon. Results from this analysis, which was carried out by other members of the consortium, indicate that elevated soil moisture across the Gangetic plains as a result of irrigation has weakened the monsooon circulation.

7.1.4 Hydrological modelling and workflow orchestration

Hydrological modelling is fundamental to the hydrological sciences. Increasingly, however, hydrology is concerned with complex problems which cannot be addressed by a single model or approach and instead require the integration of data and models from diverse sources. This gives rise to the notion of scientific workflows, which define the sequence of tasks required to process, transform and analyse data to answer complex scientific questions. Workflow orchestration is the coordination and arrangement of various tasks or services to execute a scientific workflow. Existing tools for workflow orchestration are strongly coupled to underlying modelling frameworks such as the Open Modeling Interface (Moore and Tindall, 2005; Gregersen et al., 2007) and the Object Modeling System (David et al., 2013). This means that model components must comply with the specific requirements of the modelling framework which often necessitates substantial modifications to the source code of the original model implementation (Knapen et al., 2013).

Chapter 6 argued that data processing languages, which provide sophisticated methods for processing and analysing spatial, temporal and spatio-temporal data, provide a more flexible approach to workflow orchestration. To illustrate the advantages of such an approach an experiment was devised to use R as a workflow orchestration tool for hydrological data analysis and modelling. The suitability of the R system for this purpose was enhanced through the development of a new software package, *Hydro*, which provides a set of classes and methods to represent hydrological data. Within *Hydro*, classes HydroSTF, HydroSTS and HydroSTI represent hydrological variables with different spatio-temporal geometries. They extend the corresponding *spacetime* classes STF, STS and STI to include a fourth dimension representing depth and additionally include metadata which provide a

clear description of the data using entries from the CUAHSI controlled vocabulary. Class HydroCatchment is a container class which groups together various model inputs, including those represented by HydroSTF, HydroSTS and HydroSTI, to provide a common interface to hydrological models. Two example applications of the experimental set-up, which demonstrated its utility for hydrological data processing and model coupling, respectively, highlighted several challenges and areas for improvement.

7.2 Summary of contributions to knowledge

This thesis has:

- Developed an open and extensible framework for land use change modelling which facilitates interactive model building, model intercomparison and ensemble predictions, as well as sophisticated methods for statistical analysis and model validation;
- Demonstrated the use of a dynamic land use change model for reconstructing historical land use/land cover at regional scales which assimilates available satellite data and selected biophysical and socioeconomic covariates;
- Provided further insight into the spatial and temporal variability of the rate and location of land use/land cover change in India through the development of a land use/land cover dataset for the period 1956–2010;
- Developed a methdology for estimating the rate and location of change in irrigated area of 25 crop types in India between 1956–2010, considering the biophysical suitability of grid cells to individual crops as well as changes in the spatial distribution of agricultural land uses;
- Highlighted the impact of irrigation on soil moisture in the Ganga basin, and developed two datasets with which to force climate models in order to investigate the impact of large-scale irrigation on the behaviour of the South Asian monsoon;
- Developed a more flexible way of orchestrating complex workflows in the hydrological sciences which involve data and models from diverse sources and with different spatial and temporal characteristics.

7.3 Recommendations for further work

7.3.1 Recommendations concerning the *lulcc* modelling framework

The lack of a spatio-temporal database backend to R for storing larger datasets restricts the amount of data that can be used in modelling applications because R requires all data

to be loaded into memory. SciDB is an array database system which, through the scidb4geo extension library, can efficiently handle spatio-temporal data. It can be used as a storage backend to R through the interface provided by the *scidb* package. Future work should therefore assess the feasibility of extending the analyses currently performed by *lulcc* on local data types to SciDB arrays. This would eliminate the need to load data into memory and also improve the efficiency of computations on large datasets by taking advantage of built-in support for parallelism in SciDB.

One of the primary goals of *lulcc* was to allow modellers to explore multiple model structures within the same environment. The R system itself, without any additional functionality from *lulcc*, provides state of the art methods for statistical modelling and machine learning and will continue to do so in the future as advances in these fields are made. However, there is a need to provide additional spatial allocation routines besides the CLUE, CLUE-S and ordered algorithms which the software currently includes. It would particularly benefit from the addition of deductive allocation routines because the existing routines are based on the inductive approach. Furthermore, as land use/land cover change is increasingly recognised as a process operating at multiple spatial scales, a valuable addition to the software would be a global land use change model, such as IMAGE or Nexus Land Use, to specify land use/land cover change at national or subnational administrative levels based on economic considerations. The provision of this type of land use change model would facilitate coupled applications involving nested models operating at different spatial scales.

7.3.2 Recommendations concerning future projections of land change

Future projections of land use/land cover change and agricultural intensification are necessary to constrain predictions about regional water resources and climate. In India, while the growth in agricultural land uses has largely stabilised, agricultural intensification is expected to continue (Neumann et al., 2011; Elliott et al., 2014; Haddeland et al., 2014). Given the environmental impacts of irrigation activities that have already been observed, careful planning is necessary to ensure that irrigation schemes are sustainable. Future projections should explore pathways towards sustainable agricultural development to ensure India's food and water security under various scenarios of population growth and economic development.

7.3.3 Recommendations concerning improvements to irrigated area mapping

The irrigated area dataset developed in Chapter 4 is associated with considerable uncertainty at the level of individual grid cells. To a large extent this is due to the fact that the FAO GAEZ suitability maps, which show the suitability of grid cells to irrigated crops according to prevailing biophysical conditions, are not calibrated to the actual distribution of irrigated land. Ultimately, this arises because available satellite data and processing algorithms are incapable of reliably distinguishing between irrigated and non-irrigated land or between specific crops. As remote sensing technology improves it may become possible to map the spatial distribution of major irrigated crops using satellite data, which would then facilitate the development of suitability maps based on statistical analysis.

A further improvement to the dataset described in Chapter 4 would be to incorporate information about the source of irrigation water. While district-level inventory data provide estimates of the proportion of gross irrigated area from groundwater and surfacewater, top-down methods to incorporate the source of water for irrigation may not be appropriate because inventory data do not specify the source of irrigation for individual crops, nor do they account for conjunctive use of surfacewater and groundwater. Instead, the source of irrigation could be estimated based on an understanding of the natural and manmade infrastructure that is in place to enable surfacewater and groundwater irrigation.

7.3.4 Recommendations concerning the inclusion of anthropogenic activities in Earth system modelling

A more fundamental limitation of the modelling presented in Chapter 5 is the fact that the current generation of Earth system models do not account for interactions between human activities and the water cycle. One option to incorporate irrigation activities is to dynamically couple a land surface model with a crop production model in order to improve the representation of land surface processes (e.g. Tsarouchi et al., 2014). However, in order to more accurately quantify the impact of land use/land cover change on the energy, water and carbon cycles, natural land cover and agricultural systems should be represented within a common conceptual framework (Bondeau et al., 2007; Van den Hoof et al., 2011). By extension, irrigation processes should also be included within a consistent framework in order to properly represent its effects on the various fluxes. To some extent these concerns have been addressed in recent additions to JULES, such as the inclusion of a dynamic crop growth module (Osborne et al., 2015), and a module for simulating the effects of irrigation on soil moisture. It was identified previously that one of the weaknesses of the soil moisture dataset presented here is the assumption that all grid cells have unlimited access

to water for irrigation. This assumption is also made in the new irrigation module, which means that while it can be used effectively for estimating the impact of soil moisture on the soil moisture boundary condition, it is not appropriate for quantifying the impact of irrigation on regional water resources. Developing a framework within which constraints on the availability of water resources could be built in to the model is therefore an essential step in order to properly investigate human-water interactions. This was the subject of a recent review in Hydrology and Earth Systems Science by Nazemi and Wheater (2015a,b), who present a roadmap for the future development of Earth system models considering water resources management activities.

7.3.5 Recommendations concerning the development of the R system for workflow orchestration

The experimental set-up developed in Chapter 6 has so far only been applied to two relatively straightforward case studies. Further work should focus on the development of more complex case studies in order to identify areas where the *Hydro* package in particular, and the R system in general, can be improved for the purposes of workflow orchestration and hydrological modelling. Applications should include data and models with different spatial and temporal characteristics as well as resources exposed as Web services. There are some areas for improvement that have already been elucidated. The first of these was touched upon previously: the need to develop a spatio-temporal database backend to efficiently handle large datasets. In addition, the *Hydro* classes need further refinement to provide a means to represent uncertainty as well as to make the transfer of data between tasks or services more efficient.

The purpose of the experimental set-up described in Chapter 6 was not to present R as a replacement to modelling frameworks such as OpenMI and OMS. Instead, the idea was to present a tool for orchestrating workflows involving diverse models which may include those compliant with alternative modelling frameworks. Thus, future work should investigate the possibility of developing standard methods for interfacing with existing model frameworks in order to enhance interoperability between models.

7.4 Final remarks

Understanding and quantifying the various effects of land change on the environment and society requires an integrated approach which seeks to combine models and data from different domains and with varying spatial and temporal characteristics. This thesis, guided by the demands of Hydroflux India, has largely focused on developing integrated modelling

systems for representing feedback mechanisms between anthropogenic activities and the environment at regional scales. In particular, it has focused on integrating data and models in order to generate the lower boundary condition of a climate modelling experiment which aims to further our understanding of the effect of land use/land cover change and agricultural intensification on the behaviour of the South Asian monsoon. A framework for land use change modelling as well as two intermediate datasets were developed in pursuit of this goal. Latterly, some issues around modelling complex systems were further explored through an experiment to set up the R system for workflow orchestration and model coupling. The various methods put forward in this thesis should be rigorously tested against diverse and complex case studies so that they can be refined, improved and potentially rejected in favour of alternative approaches based on new and better data and further advances in knowledge.

References

Alauddin, M. and Quiggin, J. (2008). Agricultural intensification, irrigation and the environment in South Asia: Issues and policy options. *Ecological Economics*, 65(1):111–124. doi:https://dx.doi.org/10.1016/j.ecolecon.2007.06.004.

Aldwaik, S. Z. and Pontius, R. G. (2012). Intensity analysis to unify measurements of size and stationarity of land changes by interval, category, and transition. *Landscape and Urban Planning*, 106(1):103–114. doi:https://dx.doi.org/10.1016/j.landurbplan.2012.02.010.

Allen, R. G., Pereira, L. S., Raes, D., Smith, M., et al. (1998). *Crop evapotranspiration—Guidelines for computing crop water requirements—FAO Irrigation and drainage paper 56*. Food and Agriculture Organization of the United Nations, Rome, Italy.

Altintas, I., Berkley, C., Jaeger, E., Jones, M., Ludascher, B., and Mock, S. (2004). Kepler: An extensible system for design and execution of scientific workflows. In *Proceedings of the 16th International Conference on Scientific and Statistical Database Management*, page 423–424. IEEE.

Amrith, S. S. (2016). Risk and the South Asian monsoon. *Climatic Change*. doi:https://dx.doi.org/10.1007/s10584-016-1629-x.

Annamalai, H., Hamilton, K., and Sperber, K. R. (2007). The South Asian summer monsoon and its relationship with ENSO in the IPCC AR4 simulations. *Journal of Climate*, 20(6):1071–1092. doi:https://dx.doi.org/10.1175/JCLI4035.1.

Archfield, S. A., Clark, M., Arheimer, B., Hay, L. E., McMillan, H., Kiang, J. E., Seibert, J., Hakala, K., Bock, A., Wagener, T., Farmer, W. H., Andréassian, V., Attinger, S., Viglione, A., Knight, R., Markstrom, S., and Over, T. (2015). Accelerating advances in continental domain hydrologic modeling. *Water Resources Research*, 51(12):10078–10091. doi:https://dx.doi.org/10.1002/2015WR017498.

Argent, R., Voinov, A., Maxwell, T., Cuddy, S., Rahman, J., Seaton, S., Vertessy, R., and Braddock, R. (2006). Comparing modelling frameworks – a workshop approach. *Environmental Modelling & Software*, 21(7):895–910. doi:https://dx.doi.org/10.1016/j.envsoft.2005.05.004.

Argent, R. M. (2004). An overview of model integration for environmental applications—components, frameworks and semantics. *Environmental Modelling & Software*, 19(3):219–234. doi:https://dx.doi.org/10.1016/S1364-8152(03)00150-6.

Arino, O., Bicheron, P., Achard, F., Latham, J., Witt, R., and Weber, J. (2008). The most detailed portrait of Earth. *Eur. Space Agency*, 136:25–31.

Arneth, A., Mercado, L., Kattge, J., and Booth, B. B. B. (2012). Future challenges of representing land-processes in studies on land-atmosphere interactions. *Biogeosciences*, 9(9):3587–3599. doi:https://dx.doi.org/10.5194/bg-9-3587-2012.

Ascough II, J., David, O., Krause, P., Heathman, G., Kralisch, S., Larose, M., Ahuja, L., and Kipka, H. (2012). Development and application of a modular watershed-scale hydrologic model using the Object Modeling System: Runoff response evaluation. *Transactions of the ASABE*, 55(1):117–135.

Asharaf, S., Dobler, A., and Ahrens, B. (2012). Soil Moisture–Precipitation feedback processes in the Indian summer monsoon season. *Journal of Hydrometeorology*, 13(5):1461–1474. doi:https://dx.doi.org/10.1175/JHM-D-12-06.1.

Ashfaq, M., Shi, Y., Tung, W.-w., Trapp, R. J., Gao, X., Pal, J. S., and Diffenbaugh, N. S. (2009). Suppression of South Asian summer monsoon precipitation in the 21st century. *Geophysical Research Letters*, 36(1). doi:https://dx.doi.org/10.1029/2008GL036500.

Barbosa, M. A. and Barbosa, L. S. (2009). A perspective on service orchestration. *Science of Computer Programming*, 74(9):671–687. doi:https://dx.doi.org/10.1016/j.scico.2008.09.019.

Barry, R. G. and Chorley, R. J. (2010). *Atmosphere, Weather and Climate*. Routledge, Abingdon, Ninth edition.

Bartholomé, E. and Belward, A. S. (2005). GLC2000: A new approach to global land cover mapping from Earth observation data. *International Journal of Remote Sensing*, 26(9):1959–1977. doi:https://dx.doi.org/10.1080/01431160412331291297.

Bastin, L., Cornford, D., Jones, R., Heuvelink, G. B., Pebesma, E., Stasch, C., Nativi, S., Mazzetti, P., and Williams, M. (2013). Managing uncertainty in integrated environmental modelling: The UncertWeb framework. *Environmental Modelling & Software*, 39:116–134. doi:https://dx.doi.org/10.1016/j.envsoft.2012.02.008.

Basu, B. K. (2007). Diurnal variation in precipitation over India during the summer monsoon season: Observed and model predicted. *Monthly Weather Review*, 135(6):2155–2167. doi:https://dx.doi.org/10.1175/MWR3355.1.

Beale, C. M., Lennon, J. J., Yearsley, J. M., Brewer, M. J., and Elston, D. A. (2010). Regression analysis of spatial data. *Ecology Letters*, 13(2):246–264. doi:https://dx.doi.org/10.1111/j.1461-0248.2009.01422.x.

Befort, D. J., Leckebusch, G. C., and Cubasch, U. (2016). Intraseasonal variability of the Indian summer monsoon: wet and dry events in COSMO-CLM. *Climate Dynamics*. doi:https://dx.doi.org/10.1007/s00382-016-2989-7.

Bensmann, F., Alcacer-Labrador, D., Ziegenhagen, D., and Roosmann, R. (2014). The RichWPS environment for orchestration. *ISPRS International Journal of Geo-Information*, 3(4):1334–1351. doi:https://dx.doi.org/10.3390/ijgi3041334.

Best, M. J., Pryor, M., Clark, D. B., Rooney, G. G., Essery, R. L. H., Ménard, C. B., Edwards, J. M., Hendry, M. A., Porson, A., Gedney, N., Mercado, L. M., Sitch, S., Blyth, E., Boucher, O., Cox, P. M., Grimmond, C. S. B., and Harding, R. J. (2011). The Joint UK Land Environment Simulator (JULES), model description – Part 1: Energy and water fluxes. *Geoscientific Model Development*, 4(3):677–699. doi:https://dx.doi.org/10.5194/gmd-4-677-2011.

Betrie, G., Van Griensven, A., Mohamed, Y., Popescu, I., Mynett, A., and Hummel, S. (2011). Linking SWAT and SOBEK using Open Modeling Interface (OpenMI) for sediment transport simulation in the Blue Nile River basin. *Transactions of the ASABE*, 54(5):1749–1757.

Beven, K. (1989). Changing ideas in hydrology—the case of physically-based models. *Journal of hydrology*, 105(1-2):157–172.

Beven, K. (2006). A manifesto for the equifinality thesis. *Journal of Hydrology*, 320(1-2):18–36. doi:https://dx.doi.org/10.1016/j.jhydrol.2005.07.007.

Beven, K., Buytaert, W., and Smith, L. A. (2012). On virtual observatories and modelled realities (or why discharge must be treated as a virtual variable). *Hydrological Processes*, 26(12):1905–1908. doi:https://dx.doi.org/10.1002/hyp.9261.

Beven, K. and Kirkby, M. (1979). A physically based, variable contributing area model of basin hydrology/un modèle à base physique de zone d'appel variable de l'hydrologie du bassin versant. *Hydrological Sciences Journal*, 24(1):43–69.

Beven, K. and Young, P. (2013). A guide to good practice in modeling semantics for authors and referees. *Water Resources Research*, 49(8):5092–5098. doi:https://dx.doi.org/10.1002/wrcr.20393.

Beven, K. J. (2000). Uniqueness of place and process representations in hydrological modelling. *Hydrology and Earth System Sciences Discussions*, 4(2):203–213.

Beven, K. J. (2001). Dalton medal lecture: How far can we go in distributed hydrological modelling? *Hydrology and Earth System Sciences*, 5(1):1–12.

Beven, K. J. and Cloke, H. L. (2012). Comment on "Hyperresolution global land surface modeling: Meeting a grand challenge for monitoring Earth's terrestrial water" by Eric F. Wood et al. *Water Resources Research*, 48(1). doi:https://dx.doi.org/10.1029/2011WR010982.

Bhat, G. S. (2006). The Indian drought of 2002—a sub-seasonal phenomenon? *Quarterly Journal of the Royal Meteorological Society*, 132(621):2583–2602. doi:https://dx.doi.org/10.1256/qj.05.13.

Biemans, H., Haddeland, I., Kabat, P., Ludwig, F., Hutjes, R. W. A., Heinke, J., von Bloh, W., and Gerten, D. (2011). Impact of reservoirs on river discharge and irrigation water supply during the 20th century. *Water Resources Research*, 47(3):n/a–n/a. doi:https://dx.doi.org/10.1029/2009WR008929.

Biemans, H., Speelman, L., Ludwig, F., Moors, E., Wiltshire, A., Kumar, P., Gerten, D., and Kabat, P. (2013). Future water resources for food production in five South Asian river basins and potential for adaptation—A modeling study. *Science of The Total Environment*, 468-469:S117–S131. doi:https://dx.doi.org/10.1016/j.scitotenv.2013.05.092.

Biggs, T. W., Thenkabail, P. S., Gumma, M. K., Scott, C. A., Parthasaradhi, G. R., and Turral, H. N. (2006). Irrigated area mapping in heterogeneous landscapes with MODIS time series, ground truth and census data, Krishna Basin, India. *International Journal of Remote Sensing*, 27(19):4245–4266. doi:https://dx.doi.org/10.1080/01431160600851801.

Billah, M. M., Goodall, J. L., Narayan, U., Essawy, B. T., Lakshmi, V., Rajasekar, A., and Moore, R. W. (2016). Using a data grid to automate data preparation pipelines required for regional-scale hydrologic modeling. *Environmental Modelling & Software*, 78:31–39. doi:https://dx.doi.org/10.1016/j.envsoft.2015.12.010.

Bivand, R. S., Pebesma, E., and Gomez-Rubio, V. (2013). *Applied spatial data analysis with R, Second edition*. Springer, NY.

Blair, P. and Buytaert, W. (2016). Socio-hydrological modelling: a review asking "why, what and how?". *Hydrology and Earth System Sciences*, 20(1):443–478. doi:https://dx.doi.org/10.5194/hess-20-443-2016.

Bollasina, M. A. (2014). Hydrology: Probing the monsoon pulse. *Nature Climate Change*, 4(6):422–423.

Bollasina, M. A., Ming, Y., and Ramaswamy, V. (2011). Anthropogenic aerosols and the weakening of the South Asian summer monsoon. *Science*, 334(6055):502–505.

Bonan, G. B., Levis, S., Kergoat, L., and Oleson, K. W. (2002). Landscapes as patches of plant functional types: An integrating concept for climate and ecosystem models. *Global Biogeochemical Cycles*, 16(2):5–1–5–23. doi:https://dx.doi.org/10.1029/2000GB001360.

Bondeau, A., Smith, P. C., Zaehle, S., Schaphoff, S., Lucht, W., Cramer, W., Gerten, D., Lotze-Campen, H., MüLler, C., Reichstein, M., and Smith, B. (2007). Modelling the role of agriculture for the 20th century global terrestrial carbon balance. *Global Change Biology*, 13(3):679–706. doi:https://dx.doi.org/10.1111/j.1365-2486.2006.01305.x.

Boucher, O., Myhre, G., and Myhre, A. (2004). Direct human influence of irrigation on atmospheric water vapour and climate. *Climate Dynamics*, 22(6-7). doi:https://dx.doi.org/10.1007/s00382-004-0402-4.

Bounoua, L., DeFries, R., Collatz, G., Sellers, P., and Khan, H. (2002). Effects of land cover conversion on surface climate. *Climatic Change*, 52:29–64. doi:https://dx.doi.org/10.1023/A:1013051420309.

Boysen, L. R., Brovkin, V., Arora, V. K., Cadule, P., de Noblet-Ducoudré, N., Kato, E., Pongratz, J., and Gayler, V. (2014). Global and regional effects of land-use change on climate in 21st century simulations with interactive carbon cycle. *Earth System Dynamics*, 5(2):309–319. doi:https://dx.doi.org/10.5194/esd-5-309-2014.

Bradley, A. V., Rosa, I. M., Pontius, R. G., Ahmed, S. E., Araújo, M. B., Brown, D. G., Brandão, A., Câmara, G., Carnerio, T. G., Hartley, A. J., Smith, M. J., and Ewers, R. M. (2016). SimiVal, a multi-criteria map comparison tool for land-change model projections. *Environmental Modelling & Software*, 82:229–240. doi:https://dx.doi.org/10.1016/j.envsoft.2016.04.016.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Breuer, L., Huisman, J., Willems, P., Bormann, H., Bronstert, A., Croke, B., Frede, H., Gräff, T., Hubrechts, L., Jakeman, A., Kite, G., Lanini, J., Leavesley, G., Lettenmaier, D., Lindström, G., Seibert, J., Sivapalan, M., and Viney, N. (2009). Assessing the impact of land use change on hydrology by ensemble modeling (LUCHEM). I: Model intercomparison with current land use. *Advances in Water Resources*, 32(2):129–146. doi:https://dx.doi.org/10.1016/j.advwatres.2008.10.003.

Brooks, R. and Corey, A. (1964). *Hydraulic properties of porous media*. Colorado State University, Fort Collins. Hydrological Papers 3.

Brouwer, C., Goffeau, A., and Heibloem, M. (1985). *Irrigation Water Management: Training Manual No. 1—Introduction to Irrigation*. Food and Agriculture Organization of the United Nations, Rome, Italy. http://www.fao.org/docrep/r4082e/r4082e00.htm [Accessed 22-April-2016].

Brown, D. G., Goovaerts, P., Burnicki, A., and Meng-Ying, L. (2002). Stochastic simulation of land-cover change using geostatistics and generalized additive models. *Photogrammetric Engineering & Remote Sensing*, 68(10):1051–1061.

Brown, D. G., Verburg, P. H., Pontius, R. G., and Lange, M. D. (2013). Opportunities to improve impact, integration, and evaluation of land change models. *Current Opinion in Environmental Sustainability*, 5(5):452–457. doi:https://dx.doi.org/10.1016/j.cosust.2013.07.012.

Bu, X., Yue, P., Wang, L., and Zhang, M. (2015). A scripting approach for integrating software packages and geoprocessing services into scientific workflows. In *Proceedings of the Fourth International Conference on Agro-Geoinformatics*, page 15–18. IEEE.

Buahin, C. A. and Horsburgh, J. S. (2015). Evaluating the simulation times and mass balance errors of component-based models: An application of OpenMI 2.0 to an urban stormwater system. *Environmental Modelling & Software*, 72:92–109. doi:https://dx.doi.org/10.1016/j.envsoft.2015.07.003.

Bulatewicz, T., Allen, A., Peterson, J., Staggenborg, S., Welch, S., and Steward, D. (2013). The Simple Script Wrapper for OpenMI: enabling interdisciplinary modeling studies. *Environmental Modelling & Software*, 39:283–294. doi:https://dx.doi.org/10.1016/j.envsoft.2012.07.006.

Bürger, C. M., Kollet, S., Schumacher, J., and Bösel, D. (2012). Introduction of a web service for cloud computing with the integrated hydrologic simulation platform ParFlow. *Computers & Geosciences*, 48:334–336. doi:https://dx.doi.org/10.1016/j.cageo.2012.01.007.

Buytaert, W. (2011). *topmodel: Implementation of the hydrological model TOPMODEL in R*. R package version 0.7.2-2.

Buytaert, W., Baez, S., Bustamante, M., and Dewulf, A. (2012). Web-Based environmental simulation: Bridging the gap between scientific modeling and Decision-Making. *Environmental Science & Technology*, 46(4):1971–1976. doi:https://dx.doi.org/10.1021/es2031278.

Buytaert, W., Reusser, D., Krause, S., and Renaud, J. (2008). Why can't we do better than Topmodel? *Hydrological Processes*, 22(20):4175–4179. doi:https://dx.doi.org/10.1002/hyp.7125.

Cai, Y., Judd, K. L., and Lontzek, T. S. (2012). Open science is necessary. *Nature Climate Change*, 2(5):299–299.

Câmara, G., Vinhas, L., Ferreira, K. R., De Queiroz, G. R., De Souza, R. C. M., Monteiro, A. M. V., De Carvalho, M. T., Casanova, M. A., and De Freitas, U. M. (2008). TerraLib: An open source GIS library for large-scale environmental and socio-economic applications. In *Open source approaches in spatial data handling*, pages 247–270. Springer.

Carneiro, T. G. d. S., Andrade, P. R. d., Câmara, G., Monteiro, A. M. V., and Pereira, R. R. (2013). An extensible toolbox for modeling nature–society interactions. *Environmental Modelling & Software*, 46:104–117. doi:https://dx.doi.org/10.1016/j.envsoft.2013.03.002.

Castella, J. and Verburg, P. H. (2007). Combination of process-oriented and pattern-oriented models of land-use change in a mountain area of vietnam. *Ecological Modelling*, 202(3-4):410–420. doi:https://dx.doi.org/10.1016/j.ecolmodel.2006.11.011.

Castronova, A. M. and Goodall, J. L. (2013). Simulating watersheds using loosely integrated model components: Evaluation of computational scaling using OpenMI. *Environmental Modelling & Software*, 39:304–313. doi:https://dx.doi.org/10.1016/j.envsoft.2012.01.020.

Castronova, A. M., Goodall, J. L., and Elag, M. M. (2013). Models as web services using the Open Geospatial Consortium (OGC) Web Processing Service (WPS) standard. *Environmental Modelling & Software*, 41:72–83. doi:https://dx.doi.org/10.1016/j.envsoft.2012.11.010.

Catalano, F., Alessandri, A., De Felice, M., Zhu, Z., and Myneni, R. B. (2016). Observationally based analysis of land–atmosphere coupling. *Earth System Dynamics*, 7(1):251–266. doi:https://dx.doi.org/10.5194/esd-7-251-2016.

Chambers, J. (2008). *Software for data analysis: programming with R*. Springer.

Chambers, J. M. (1998). *Programming with data: a guide to the S language*. Springer, New York.

Chambers, J. M. (2000). Users, Programmers, and Statistical Software. *Journal of Computational and Graphical Statistics*, 9(3):404–422. doi:https://dx.doi.org/10.1080/10618600.2000.10474890.

Chen, J., Chen, J., Liao, A., Cao, X., Chen, L., Chen, X., He, C., Han, G., Peng, S., Lu, M., Zhang, W., Tong, X., and Mills, J. (2015). Global land cover mapping at 30m resolution: A POK-based operational approach. *ISPRS Journal of Photogrammetry and Remote Sensing*, 103:7–27. doi:https://dx.doi.org/10.1016/j.isprsjprs.2014.09.002.

Chou, C. (2003). Land-sea heating contrast in an idealized Asian summer monsoon. *Climate Dynamics*, 21(1):11–25. doi:https://dx.doi.org/10.1007/s00382-003-0315-7.

Claes, M., Mens, T., and Grosjean, P. (2014). On the maintainability of CRAN packages. In *Software Maintenance, Reengineering and Reverse Engineering (CSMR-WCRE), 2014 Software Evolution Week-IEEE Conference on*, page 308–312. IEEE.

Clark, D. B., Mercado, L. M., Sitch, S., Jones, C. D., Gedney, N., Best, M. J., Pryor, M., Rooney, G. G., Essery, R. L. H., Blyth, E., Boucher, O., Harding, R. J., Huntingford, C., and Cox, P. M. (2011a). The Joint UK Land Environment Simulator (JULES), model description – Part 2: Carbon fluxes and vegetation dynamics. *Geoscientific Model Development*, 4(3):701–722. doi:https://dx.doi.org/10.5194/gmd-4-701-2011.

Clark, M. P., Fan, Y., Lawrence, D. M., Adam, J. C., Bolster, D., Gochis, D. J., Hooper, R. P., Kumar, M., Leung, L. R., Mackay, D. S., Maxwell, R. M., Shen, C., Swenson, S. C., and Zeng, X. (2015a). Improving the representation of hydrologic processes in earth system models. *Water Resources Research*, pages n/a–n/a. doi:https://dx.doi.org/10.1002/2015WR017096.

Clark, M. P., Kavetski, D., and Fenicia, F. (2011b). Pursuing the method of multiple working hypotheses for hydrological modeling. *Water Resources Research*, 47(9):n/a–n/a. doi:https://dx.doi.org/10.1029/2010WR009827.

Clark, M. P., Kavetski, D., and Fenicia, F. (2012). Reply to comment by K. Beven et al. on "Pursuing the method of multiple working hypotheses for hydrological modeling". *Water Resources Research*, 48(11):n/a–n/a. doi:https://dx.doi.org/10.1029/2012WR012547.

Clark, M. P., Nijssen, B., Lundquist, J. D., Kavetski, D., Rupp, D. E., Woods, R. A., Freer, J. E., Gutmann, E. D., Wood, A. W., Brekke, L. D., Arnold, J. R., Gochis, D. J., and Rasmussen, R. M. (2015b). A unified approach for process-based hydrologic modeling: 1. Modeling concept. *Water Resources Research*, 51(4):2498–2514. doi:https://dx.doi.org/10.1002/2015WR017198.

Clark, M. P., Nijssen, B., Lundquist, J. D., Kavetski, D., Rupp, D. E., Woods, R. A., Freer, J. E., Gutmann, E. D., Wood, A. W., Gochis, D. J., Rasmussen, R. M., Tarboton, D. G., Mahat, V., Flerchinger, G. N., and Marks, D. G. (2015c). A unified approach for process-based hydrologic modeling: 2. Model implementation and case studies. *Water Resources Research*, 51(4):2515–2542. doi:https://dx.doi.org/10.1002/2015WR017200.

Clark, M. P., Schaefli, B., Schymanski, S. J., Samaniego, L., Luce, C. H., Jackson, B. M., Freer, J. E., Arnold, J. R., Moore, R. D., Istanbulluoglu, E., and Ceola, S. (2016). Improving the theoretical underpinnings of process-based hydrologic models. *Water Resources Research*, 52(3):2350–2365. doi:https://dx.doi.org/10.1002/2015WR017910.

Clark, M. P., Slater, A. G., Rupp, D. E., Woods, R. A., Vrugt, J. A., Gupta, H. V., Wagener, T., and Hay, L. E. (2008). Framework for Understanding Structural Errors (FUSE): A modular framework to diagnose differences between hydrological models. *Water Resources Research*, 44(12):n/a–n/a. doi:https://dx.doi.org/10.1029/2007WR006735.

Cook, B. I., Puma, M. J., and Krakauer, N. Y. (2011). Irrigation induced surface cooling in the context of modern and increased greenhouse gas forcing. *Climate Dynamics*, 37(7-8):1587–1600. doi:https://dx.doi.org/10.1007/s00382-010-0932-x.

Cox, P., Huntingford, C., and Harding, R. (1998). A canopy conductance and photosynthesis model for use in a GCM land surface scheme. *Journal of Hydrology*, 212–213:79–94. doi:https://dx.doi.org/10.1016/S0022-1694(98)00203-0.

Dash, S. K., Jenamani, R. K., Kalsi, S. R., and Panda, S. K. (2007). Some evidence of climate change in twentieth-century India. *Climatic Change*, 85(3-4):299–321. doi:https://dx.doi.org/10.1007/s10584-007-9305-9.

David, O., Ascough, J., Lloyd, W., Green, T., Rojas, K., Leavesley, G., and Ahuja, L. (2013). A software engineering perspective on environmental modeling framework design: The Object Modeling System. *Environmental Modelling & Software*, 39:201–213. doi:https://dx.doi.org/10.1016/j.envsoft.2012.03.006.

David, O., Ascough II, J., Leavesley, G., and Ahuja, L. (2010). Rethinking modeling framework design: Object Modeling System 3.0. In Swayne, D. A., Yang, W., Voinov, A., Rizzoli, A., and Filatova, T., editors, *International Environmental Modelling and Software Society (iEMSs), 2010 International Congress on Environmental Modelling and Software, Modelling for Environment's Sake, Fifth Biennial Meeting, Ottowa, Canada*, pages 1–9.

David, O., Markstrom, S. L., Rojas, K. W., Ahuja, L. R., and Schneider, I. W. (2002). The Object Modeling System. In Ahuja, L. R., Ma, L., and Howell, T. A., editors, *Agricultural system models in field research and technology transfer*, pages 317–331. CRC Press, FL USA.

Dheeravath, V., Thenkabail, P., Chandrakantha, G., Noojipady, P., Reddy, G., Biradar, C., Gumma, M., and Velpuri, M. (2010). Irrigated areas of India derived using MODIS 500m time series for the years 2001–2003. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(1):42–59. doi:https://dx.doi.org/10.1016/j.isprsjprs.2009.08.004.

Di Baldassarre, G., Brandimarte, L., and Beven, K. (2015). The seventh facet of uncertainty: wrong assumptions, unknowns and surprises in the dynamics of human-water systems. *Hydrological Sciences Journal*, page 150911074729008. doi:https://dx.doi.org/10.1080/02626667.2015.1091460.

Dirmeyer, P., Koster, R., and Guo, Z. (2006). Do global models properly represent the feedback between land and atmosphere? *Journal of Hydrometeorology*, 7:1177–1198.

Dirmeyer, P. A., Schlosser, C. A., and Brubaker, K. L. (2009). Precipitation, recycling, and land memory: An integrated analysis. *Journal of Hydrometeorology*, 10(1):278–288. doi:https://dx.doi.org/10.1175/2008JHM1016.1.

Döll, P., Hoffmann-Dobrev, H., Portmann, F., Siebert, S., Eicker, A., Rodell, M., Strassberg, G., and Scanlon, B. (2012). Impact of water withdrawals from groundwater and surface water on continental water storage variations. *Journal of Geodynamics*, 59-60:143–156. doi:https://dx.doi.org/10.1016/j.jog.2011.05.001.

Döll, P., Müller Schmied, H., Schuh, C., Portmann, F. T., and Eicker, A. (2014). Global-scale assessment of groundwater depletion and related groundwater abstractions: Combining hydrological modeling with information from well observations and GRACE satellites. *Water Resources Research*, 50(7):5698–5720. doi:https://dx.doi.org/10.1002/2014WR015595.

Döll, P. and Siebert, S. (2002). Global modeling of irrigation water requirements. *Water Resources Research*, 38(4):8–1–8–10. doi:https://dx.doi.org/10.1029/2001WR000355.

Döll, P., Siebert, S., et al. (2000). A digital global map of irrigated areas. *Icid Journal*, 49(2):55–66.

Douglas, E. M., Niyogi, D., Frolking, S., Yeluripati, J. B., Pielke, R. A., Niyogi, N., Vörösmarty, C. J., and Mohanty, U. C. (2006). Changes in moisture and energy fluxes due to agricultural land use and irrigation in the Indian Monsoon Belt. *Geophysical Research Letters*, 33(14). doi:https://dx.doi.org/10.1029/2006GL026550.

Douville, H., Chauvin, F., and Broqua, H. (2001). Influence of soil moisture on the Asian and African monsoons. Part I: Mean monsoon and daily precipitation. *Journal of Climate*, 14(11):2381–2403.

Dozier, A., David, O., Arabi, M., Lloyd, W., and Zhang, Y. (2016). A minimally invasive model data passing interface for integrating legacy environmental system models. *Environmental Modelling & Software*, 80:265–280. doi:https://dx.doi.org/10.1016/j.envsoft.2016.02.031.

Duan, Q., Schaake, J., Andréassian, V., Franks, S., Goteti, G., Gupta, H., Gusev, Y., Habets, F., Hall, A., Hay, L., Hogue, T., Huang, M., Leavesley, G., Liang, X., Nasonova, O., Noilhan, J., Oudin, L., Sorooshian, S., Wagener, T., and Wood, E. (2006). Model parameter estimation experiment (MOPEX): an overview of science strategy and major results from the second and third workshops. *Journal of Hydrology*, 320(1-2):3–17. doi:https://dx.doi.org/10.1016/j.jhydrol.2005.07.031.

Dunn, S. M., Freer, J., Weiler, M., Kirkby, M. J., Seibert, J., Quinn, P. F., Lischeid, G., Tetzlaff, D., and Soulsby, C. (2008). Conceptualization in catchment modelling: simply learning? *Hydrological Processes*, 22(13):2389–2393. doi:https://dx.doi.org/10.1002/hyp.7070.

Echeverria, C., Coomes, D. A., Hall, M., and Newton, A. C. (2008). Spatially explicit models to analyze forest loss and fragmentation between 1976 and 2020 in southern Chile. *Ecological Modelling*, 212(3-4):439–449. doi:https://dx.doi.org/10.1016/j.ecolmodel.2007.10.045.

Elliott, J., Deryng, D., Muller, C., Frieler, K., Konzmann, M., Gerten, D., Glotter, M., Florke, M., Wada, Y., Best, N., Eisner, S., Fekete, B. M., Folberth, C., Foster, I., Gosling, S. N., Haddeland, I., Khabarov, N., Ludwig, F., Masaki, Y., Olin, S., Rosenzweig, C., Ruane, A. C., Satoh, Y., Schmid, E., Stacke, T., Tang, Q., and Wisser, D. (2014). Constraints and potentials of future irrigation water availability on agricultural production under climate change. *Proceedings of the National Academy of Sciences*, 111(9):3239–3244. doi:https://dx.doi.org/10.1073/pnas.1222474110.

Elshafei, Y., Sivapalan, M., Tonts, M., and Hipsey, M. R. (2014). A prototype framework for models of socio-hydrology: identification of key feedback loops and parameterisation approach. *Hydrology and Earth System Sciences*, 18(6):2141–2166. doi:https://dx.doi.org/10.5194/hess-18-2141-2014.

Essery, R. and Clark, D. B. (2003). Developments in the MOSES 2 land-surface model for PILPS 2e. *Global and Planetary Change*, 38(1-2):161–164. doi:https://dx.doi.org/10.1016/S0921-8181(03)00026-2.

Evenson, R. E. and Gollin, D. (2003). Assessing the impact of the Green Revolution, 1960 to 2000. *Science*, 300(5620):758–762.

FAO (2015a). AQUASTAT. Available from: http://www.fao.org/nr/water/aquastat/main/index.stm [Accessed 04-December-2015].

FAO (2015b). FAOSTAT. Available from: http://faostat3.fao.org/home/E [Accessed 25-November-2015].

FAO (2015c). World Programme for the Census of Agriculture. Available from: http://www.fao.org/economic/ess/ess-home/en/ [Accessed 05-December-2015].

FAO/IIASA/ISRIC/ISS-CAS/JRC (2012). Harmonized World Soil Database v1.2. Available from: http://www.fao.org/soils-portal/soil-survey/soil-maps-and-databases/harmonized-world-soil-database-v12/en/ [Accessed 01-August-2015].

Fatichi, S., Vivoni, E. R., Ogden, F. L., Ivanov, V. Y., Mirus, B., Gochis, D., Downer, C. W., Camporese, M., Davison, J. H., Ebel, B., Jones, N., Kim, J., Mascaro, G., Niswonger, R., Restrepo, P., Rigon, R., Shen, C., Sulis, M., and Tarboton, D. (2016). An overview of current applications, challenges, and future trends in distributed process-based models in hydrology. *Journal of Hydrology*, 537:45–60. doi:https://dx.doi.org/10.1016/j.jhydrol.2016.03.026.

Feddema, J. J. (2005). The importance of Land-Cover change in simulating future climates. *Science*, 310(5754):1674–1678. doi:https://dx.doi.org/10.1126/science.1118160.

Fenicia, F., Kavetski, D., and Savenije, H. H. G. (2011). Elements of a flexible approach for conceptual hydrological modeling: 1. Motivation and theoretical development. *Water Resources Research*, 47(11):n/a–n/a. doi:https://dx.doi.org/10.1029/2010WR010174.

Fenicia, F., Savenije, H. H. G., Matgen, P., and Pfister, L. (2008). Understanding catchment behavior through stepwise model concept improvement. *Water Resources Research*, 44(1):n/a–n/a. doi:https://dx.doi.org/10.1029/2006WR005563.

Fiske, I. and Chandler, R. (2011). unmarked: An R package for fitting hierarchical models of wildlife occurrence and abundance. *Journal of Statistical Software*, 43(10):1–23.

Fitzgerald, S., Smith, T., and Anashua, M. (2016). R code for three map comparison at multiple resolutions [video]. Available from: https://www.youtube.com/watch?v=F2qg5YmjHg4 [Accessed 09-June-2016].

Foley, J. A. (2005). Global consequences of land use. *Science*, 309(5734):570–574. doi:https://dx.doi.org/10.1126/science.1111772.

Foley, J. A., Ramankutty, N., Brauman, K. A., Cassidy, E. S., Gerber, J. S., Johnston, M., Mueller, N. D., O'Connell, C., Ray, D. K., West, P. C., Balzer, C., Bennett, E. M., Carpenter, S. R., Hill, J., Monfreda, C., Polasky, S., Rockström, J., Sheehan, J., Siebert, S., Tilman, D., and Zaks, D. P. M. (2011). Solutions for a cultivated planet. *Nature*, 478(7369):337–342. doi:https://dx.doi.org/10.1038/nature10452.

Formetta, G., Antonello, A., Franceschi, S., David, O., and Rigon, R. (2014). Hydrological modelling with components: A GIS-based open-source framework. *Environmental Modelling & Software*, 55:190–200. doi:https://dx.doi.org/10.1016/j.envsoft.2014.01.019.

Formetta, G., Capparelli, G., David, O., Green, T., and Rigon, R. (2016). Integration of a three-dimensional process-based hydrological model into the Object Modeling System. *Water*, 8(1):12. doi:https://dx.doi.org/10.3390/w8010012.

Fraiture, C. d., Giordano, M., and Liao, Y. (2008). Biofuels and implications for agricultural water use: blue impacts of green energy. *Water Policy*, 10(S1):67. doi:https://dx.doi.org/10.2166/wp.2008.054.

Freydank, K. and Siebert, S. (2008). *Towards mapping the extent of irrigation in the last century: time series of irrigated area per country*. Institute of Physical Geography, University of Frankfurt, Frankfurt, Germany. Frankfurt Hydrology Paper No. 8.

Friedl, M. A., McIver, D. K., Hodges, J. C., Zhang, X. Y., Muchoney, D., Strahler, A. H., Woodcock, C. E., Gopal, S., Schneider, A., Cooper, A., et al. (2002). Global land cover mapping from MODIS: algorithms and early results. *Remote Sensing of Environment*, 83(1):287–302.

Friedl, M. A., Sulla-Menashe, D., Tan, B., Schneider, A., Ramankutty, N., Sibley, A., and Huang, X. (2010). MODIS collection 5 global land cover: Algorithm refinements and characterization of new datasets. *Remote Sensing of Environment*, 114(1):168–182. doi:https://dx.doi.org/10.1016/j.rse.2009.08.016.

Fuchs, R., Herold, M., Verburg, P. H., Clevers, J. G., and Eberle, J. (2015). Gross changes in reconstructions of historic land cover/use for Europe between 1900 and 2010. *Global Change Biology*, 21(1):299–313. doi:https://dx.doi.org/10.1111/gcb.12714.

Fuchs, R., Herold, M., Verburg, P. H., and Clevers, J. G. P. W. (2013). A high-resolution and harmonized model approach for reconstructing and analysing historic land changes in Europe. *Biogeosciences*, 10(3):1543–1559. doi:https://dx.doi.org/10.5194/bg-10-1543-2013.

Gadgil, S. (2003). The Indian monsoon and its variability. *Annual Review of Earth and Planetary Sciences*, 31(1):429–467. doi:https://dx.doi.org/10.1146/annurev.earth.31.100901.141251.

GADM (2015). Global Administrative Areas Database. Available from: http://www.gadm.org/ [Accessed 01-August-2015].

Garcia, M., Portney, K., and Islam, S. (2016). A question driven socio-hydrological modeling process. *Hydrology and Earth System Sciences*, 20(1):73–92. doi:https://dx.doi.org/10.5194/hess-20-73-2016.

Gebbert, S. and Pebesma, E. (2014). A temporal GIS for field based environmental modeling. *Environmental Modelling & Software*, 53:1–12. doi:https://dx.doi.org/10.1016/j.envsoft.2013.11.001.

Geller, G. N. and Melton, F. (2008). Looking forward: Applying an ecological model web to assess impacts of climate change. *Biodiversity*, 9(3-4):79–83. doi:https://dx.doi.org/10.1080/14888386.2008.9712910.

Geller, G. N. and Turner, W. (2007). The model web: a concept for ecological forecasting. In *Geoscience and Remote Sensing Symposium, 2007. IGARSS 2007. IEEE International*, page 2469–2472. IEEE.

Gerten, D., Rost, S., von Bloh, W., and Lucht, W. (2008). Causes of change in 20th century global river discharge. *Geophysical Research Letters*, 35(20). doi:https://dx.doi.org/10.1029/2008GL035258.

Ghosh, S., Das, D., Kao, S., and Ganguly, A. R. (2011). Lack of uniform trends but increasing spatial variability in observed Indian rainfall extremes. *Nature Climate Change*, 2(2):86–91. doi:https://dx.doi.org/10.1038/nclimate1327.

Gil, Y., Deelman, E., Ellisman, M., Fahringer, T., Fox, G., Gannon, D., Goble, C., Livny, M., Moreau, L., and Myers, J. (2007). Examining the challenges of scientific workflows. *IEEE Computer*, 40(12):26–34.

Gleeson, T., Wada, Y., Bierkens, M. F. P., and van Beek, L. P. H. (2012). Water balance of global aquifers revealed by groundwater footprint. *Nature*, 488(7410):197–200. doi:https://dx.doi.org/10.1038/nature11295.

Gober, P. and Wheater, H. S. (2015). Debates-Perspectives on socio-hydrology: Modeling flood risk as a public policy problem. *Water Resources Research*, 51(6):4782–4788. doi:https://dx.doi.org/10.1002/2015WR016945.

Goodall, J. L., Castronova, A. M., Huynh, N. N., Caicedo, J. M., and Hall, R. (2013). Using a Service-Oriented approach to simulate integrated urban infrastructure systems. *Journal of Computing in Civil Engineering*, 29(5):04014061.

Goodall, J. L., Robinson, B. F., and Castronova, A. M. (2011). Modeling water resource systems using a service-oriented computing paradigm. *Environmental Modelling & Software*, 26(5):573–582. doi:https://dx.doi.org/10.1016/j.envsoft.2010.11.013.

Görlach, K., Sonntag, M., Karastoyanova, D., Leymann, F., and Reiter, M. (2011). Conventional workflow technology for scientific simulation. In Yang, X., Wang, L., and Jie, W., editors, *Guide to e-Science*, pages 323–352. Springer London, London.

Goswami, B. N., Wu, G., and Yasunari, T. (2006). The annual cycle, intraseasonal oscillations, and the roadblock to seasonal predictability of the Asian summer monsoon. *Journal of climate*, 19(20):5078–5099.

Green, T. R., Erskine, R. H., Coleman, M. L., David, O., Ascough, J. C., and Kipka, H. (2015). The AgroEcoSystem (AgES) response-function model simulates layered soil-water dynamics in semiarid Colorado: Sensitivity and calibration. *Vadose Zone Journal*, 14(8).

Gregersen, J. B., Gijsbers, P. J. A., and Westen, S. J. P. (2007). OpenMI: open modelling interface. *Journal of Hydroinformatics*, 9(3):175. doi:https://dx.doi.org/10.2166/hydro.2007.023.

Guo, Z., Dirmeyer, P. A., Koster, R. D., Sud, Y. C., Bonan, G., Oleson, K. W., Chan, E., Verseghy, D., Cox, P., Gordon, C. T., et al. (2006). GLACE: the global land-atmosphere coupling experiment. Part II: analysis. *Journal of Hydrometeorology*, 7(4):611–625.

Gupta, H. V. and Nearing, G. S. (2014). Debates-the future of hydrological sciences: A (common) path forward? using models and data to learn: A systems theoretic perspective on the future of hydrological science. *Water Resources Research*, 50(6):5351–5359. doi:https://dx.doi.org/10.1002/2013WR015096.

Gupta, T., Jonesb, R., Bastinb, L., and Cornfordb, D. (2012). Integrating OpenMI and UncertWeb: managing uncertainty in OpenMI models. In *Managing Resources of a Limited Planet: Pathways and Visions under Uncertainty, IEMSS Sixth International Congress on Environmental Modelling and Software*, page 1–5.

Gururaja, K. V. and Sudhira, H. S. (2012). Population crunch in India: is it urban or still rural? *Current Science*, 103(1):37–40.

Haddeland, I., Heinke, J., Biemans, H., Eisner, S., Flörke, M., Hanasaki, N., Konzmann, M., Ludwig, F., Masaki, Y., Schewe, J., Stacke, T., Tessler, Z. D., Wada, Y., and Wisser, D. (2014). Global water resources affected by human interventions and climate change. *Proceedings of the National Academy of Sciences*, 111(9):3251–3256. doi:https://dx.doi.org/10.1073/pnas.1222475110.

Haddeland, I., Lettenmaier, D. P., and Skaugen, T. (2006a). Effects of irrigation on the water and energy balances of the Colorado and Mekong river basins. *Journal of Hydrology*, 324(1-4):210–223. doi:https://dx.doi.org/10.1016/j.jhydrol.2005.09.028.

Haddeland, I., Skaugen, T., and Lettenmaier, D. P. (2006b). Anthropogenic impacts on continental surface water fluxes. *Geophysical Research Letters*, 33(8). doi:https://dx.doi.org/10.1029/2006GL026047.

Haddeland, I., Skaugen, T., and Lettenmaier, D. P. (2007). Hydrologic effects of land and water management in North America and Asia: 1700–1992. *Hydrology and Earth System Sciences Discussions*, 11(2):1035–1045.

Hanjra, M. A. and Qureshi, M. E. (2010). Global water crisis and future food security in an era of climate change. *Food Policy*, 35(5):365–377. doi:https://dx.doi.org/10.1016/j.foodpol.2010.05.006.

Hansen, M. C., Defries, R. S., Townshend, J. R. G., and Sohlberg, R. (2000). Global land cover classification at 1 km spatial resolution using a classification tree approach. *International Journal of Remote Sensing*, 21(6-7):1331–1364. doi:https://dx.doi.org/10.1080/014311600210209.

Harding, K. J., Twine, T. E., and Lu, Y. (2015). Effects of dynamic crop growth on the simulated precipitation response to irrigation. *Earth Interactions*, 19(14):1–31.

Hasson, S., Lucarini, V., and Pascale, S. (2013). Hydrological cycle over South and Southeast Asian river basins as simulated by PCMDI/CMIP3 experiments. *Earth System Dynamics*, 4(2):199–217. doi:https://dx.doi.org/10.5194/esd-4-199-2013.

Hengl, T., de Jesus, J. M., MacMillan, R. A., Batjes, N. H., Heuvelink, G. B. M., Ribeiro, E., Samuel-Rosa, A., Kempen, B., Leenaars, J. G. B., Walsh, M. G., and Gonzalez, M. R. (2014). SoilGrids1km âĂŤ Global Soil Information Based on Automated Mapping. *PLoS ONE*, 9(8):e105992.

Hewitt, R., Díaz Pacheco, J., and Moya Gómez, B. (2013). A cellular automata land use model for the R software environment. Available from: https://simlander.wordpress.com/ [Accessed 11-January-2015].

Hijmans, R. J. (2014). *raster: Geographic data analysis and modeling*. R package version 2.2-31.

Hill, C., DeLuca, C., Suarez, M., Da Silva, A., et al. (2004). The architecture of the Earth System Modeling Framework. *Computing in Science & Engineering*, 6(1):18–28.

Hillel, D. (1998). *Environmental Soil Physics*. Academic Press, London.

Hinz, M., Nüst, D., Proß, B., and Pebesma, E. (2013). Spatial Statistics on the Geospatial Web. In *The 16th AGILE International Conference on Geographic Information Science, Short Papers*, pages 1–7.

Hira, G. (2009). Water management in northern states and the food security of India. *Journal of Crop Improvement*, 23(2):136–157.

Horsburgh, J. S., Tarboton, D. G., Hooper, R. P., and Zaslavsky, I. (2014). Managing a community shared vocabulary for hydrologic observations. *Environmental Modelling & Software*, 52:62–73. doi:https://dx.doi.org/10.1016/j.envsoft.2013.10.012.

Houghton, R. A., House, J. I., Pongratz, J., van der Werf, G. R., DeFries, R. S., Hansen, M. C., Le Quéré, C., and Ramankutty, N. (2012). Carbon emissions from land use and land-cover change. *Biogeosciences*, 9(12):5125–5142. doi:https://dx.doi.org/10.5194/bg-9-5125-2012.

Hrachowitz, M., Savenije, H., Blöschl, G., McDonnell, J., Sivapalan, M., Pomeroy, J., Arheimer, B., Blume, T., Clark, M., Ehret, U., Fenicia, F., Freer, J., Gelfan, A., Gupta, H., Hughes, D., Hut, R., Montanari, A., Pande, S., Tetzlaff, D., Troch, P., Uhlenbrook, S., Wagener, T., Winsemius, H., Woods, R., Zehe, E., and Cudennec, C. (2013). A decade of predictions in ungauged basins (PUB)—a review. *Hydrological Sciences Journal*, 58(6):1198–1255. doi:https://dx.doi.org/10.1080/02626667.2013.803183.

Humphreys, E., Kukal, S. S., Christen, E. W., Hira, G. S., Sharma, R. K., et al. (2010). 5 halting the groundwater decline in North-West India—Which crop technologies will be winners? *Advances in Agronomy*, 109(5):155–217.

Icrisat (2016). Village dynamics in South Asia. Available from: http://vdsa.icrisat.ac.in/ [Accessed 01-August-2015].

IIASA, F. (2012). *Global Agro-ecological Zones (GAEZ v3.0)*. International Institute for Applied Systems Analysis, Laxenburg, Austria and Food and Agriculture Organization of the United Nations, Rome, Italy.

Ince, D. C., Hatton, L., and Graham-Cumming, J. (2012). The case for open computer programs. *Nature*, 482(7386):485–488. doi:https://dx.doi.org/10.1038/nature10836.

Indiastat (2016). Socio-economic statistical information about India. Available from: http://www.indiastat.com/default.aspx [Accessed 01-August-2015].

James, K. S. (2011). India's demographic change: Opportunities and challenges. *Science*, 333(6042):576–580. doi:https://dx.doi.org/10.1126/science.1207969.

Jha, S., Katz, D. S., Luckow, A., Merzky, A., and Stamou, K. (2011). Understanding scientific applications for cloud environments. *Cloud Computing: Principles and Paradigms*, page 345–371.

Joppa, L. N., McInerny, G., Harper, R., Salido, L., Takeda, K., O'Hara, K., Gavaghan, D., and Emmott, S. (2013). Troubling trends in scientific software use. *Science*, 340(6134):814–815.

Ju, J. and Roy, D. P. (2008). The availability of cloud-free Landsat ETM+ data over the conterminous United States and globally. *Remote Sensing of Environment*, 112(3):1196–1211.

Kadlec, J., StClair, B., Ames, D. P., and Gill, R. A. (2015). WaterML R package for managing ecological experiment data on a CUAHSI HydroServer. *Ecological Informatics*, 28:19–28. doi:https://dx.doi.org/10.1016/j.ecoinf.2015.05.002.

Kampf, S. K. and Burges, S. J. (2007). A framework for classifying and comparing distributed hillslope and catchment hydrologic models. *Water Resources Research*, 43(5):n/a–n/a. doi:https://dx.doi.org/10.1029/2006WR005370.

Kavetski, D. and Fenicia, F. (2011). Elements of a flexible approach for conceptual hydrological modeling: 2. Application and experimental insights. *Water Resources Research*, 47(11):n/a–n/a. doi:https://dx.doi.org/10.1029/2011WR010748.

Kavetski, D., Kuczera, G., and Franks, S. W. (2006). Calibration of conceptual hydrological models revisited: 1. Overcoming numerical artefacts. *Journal of Hydrology*, 320(1-2):173–186. doi:https://dx.doi.org/10.1016/j.jhydrol.2005.07.012.

Kelly (Letcher), R. A., Jakeman, A. J., Barreteau, O., Borsuk, M. E., ElSawah, S., Hamilton, S. H., Henriksen, H. J., Kuikka, S., Maier, H. R., Rizzoli, A. E., van Delden, H., and Voinov, A. A. (2013). Selecting among five common modelling approaches for integrated environmental assessment and management. *Environmental Modelling & Software*, 47:159–181. doi:https://dx.doi.org/10.1016/j.envsoft.2013.05.005.

Kirchner, J. W. (2006). Getting the right answers for the right reasons: Linking measurements, analyses, and models to advance the science of hydrology. *Water Resources Research*, 42(3):n/a–n/a. doi:https://dx.doi.org/10.1029/2005WR004362.

Kitoh, A., Endo, H., Krishna Kumar, K., Cavalcanti, I. F. A., Goswami, P., and Zhou, T. (2013). Monsoons in a changing world: A regional perspective in a global context. *Journal of Geophysical Research: Atmospheres*, 118(8):3053–3065. doi:https://dx.doi.org/10.1002/jgrd.50258.

Klein Goldewijk, K., Beusen, A., and Janssen, P. (2010). Long-term dynamic modeling of global population and built-up area in a spatially explicit way: HYDE 3.1. *The Holocene*. doi:https://dx.doi.org/10.1177/0959683609356587.

Klein Goldewijk, K., Beusen, A., Van Drecht, G., and De Vos, M. (2011). The HYDE 3.1 spatially explicit database of human-induced global land-use change over the past 12,000 years. *Global Ecology and Biogeography*, 20(1):73–86.

Knapen, R., Janssen, S., Roosenschoon, O., Verweij, P., de Winter, W., Uiterwijk, M., and Wien, J. (2013). Evaluating OpenMI as a model integration platform across disciplines. *Environmental Modelling & Software*, 39:274–282. doi:https://dx.doi.org/10.1016/j.envsoft.2012.06.011.

Knutson, T. R., Delworth, T. L., Dixon, K. W., Held, I. M., Lu, J., Ramaswamy, V., Schwarzkopf, M. D., Stenchikov, G., and Stouffer, R. J. (2006). Assessment of twentieth-century regional surface temperature trends using the GFDL CM2 coupled models. *Journal of Climate*, 19(9):1624–1651.

Knutti, R. and Sedláček, J. (2012). Robustness and uncertainties in the new CMIP5 climate model projections. *Nature Climate Change*, 3(4):369–373. doi:https://dx.doi.org/10.1038/nclimate1716.

Koster, R. D., Dirmeyer, P. A., Guo, Z., Bonan, G., Chan, E., Cox, P., Gordon, C. T., Kanae, S., Kowalczyk, E., Lawrence, D., Liu, P., Lu, C.-H., Malyshev, S., McAvaney, B., Mitchell, K., Mocko, D., Oki, T., Oleson, K., Pitman, A., Sud, Y. C., Taylor, C. M., Verseghy, D., Vasic, R., Xue, Y., and Yamada, T. (2004). Regions of strong coupling between soil moisture and precipitation. *Science*, 305(5687):1138–1140. doi:https://dx.doi.org/10.1126/science.1100217.

Koster, R. D., Guo, Z., Yang, R., Dirmeyer, P. A., Mitchell, K., and Puma, M. J. (2009). On the nature of soil moisture in land surface models. *Journal of Climate*, 22(16):4322–4335. doi:https://dx.doi.org/10.1175/2009JCLI2832.1.

Koster, R. D., Sud, Y. C., Guo, Z., Dirmeyer, P. A., Bonan, G., Oleson, K. W., Chan, E., Verseghy, D., Cox, P., Davies, H., et al. (2006). GLACE: the global land-atmosphere coupling experiment. Part I: Overview. *Journal of Hydrometeorology*, 7(4):590–610.

Krishna Kumar, K., Rupa Kumar, K., Ashrit, R. G., Deshpande, N. R., and Hansen, J. W. (2004). Climate impacts on Indian agriculture. *International Journal of Climatology*, 24(11):1375–1393. doi:https://dx.doi.org/10.1002/joc.1081.

Krishnamurthy, V. and Shukla, J. (2007). Intraseasonal and seasonally persisting patterns of Indian monsoon rainfall. *Journal of Climate*, 20(1):3–20. doi:https://dx.doi.org/10.1175/JCLI3981.1.

Krishnan, R., Sabin, T., Ayantika, D., Kitoh, A., Sugi, M., Murakami, H., Turner, A., Slingo, J., and Rajendran, K. (2013). Will the South Asian monsoon overturning circulation stabilize any further? *Climate dynamics*, 40(1-2):187–211.

Krishnan, R., Sabin, T. P., Vellore, R., Mujumdar, M., Sanjay, J., Goswami, B. N., Hourdin, F., Dufresne, J., and Terray, P. (2015). Deciphering the desiccation trend of the South Asian monsoon hydroclimate in a warming world. *Climate Dynamics*. doi:https://dx.doi.org/10.1007/s00382-015-2886-5.

Kumar, P. (2015). Hydrocomplexity: Addressing water security and emergent environmental risks. *Water Resources Research*, 51(7):5827–5838. doi:https://dx.doi.org/10.1002/2015WR017342.

Kunnath-Poovakka, A., Ryu, D., Renzullo, L., and George, B. (2016). The efficacy of calibrating hydrologic model using remotely sensed evapotranspiration and soil moisture for streamflow prediction. *Journal of Hydrology*, 535:509–524.

Lall, U. (2014). Debates-The future of hydrological sciences: A (common) path forward? one water. one world. many climes. many souls. *Water Resources Research*, 50(6):5335–5341. doi:https://dx.doi.org/10.1002/2014WR015402.

Lambin, E. F. and Meyfroidt, P. (2011). Global land use change, economic globalization, and the looming land scarcity. *Proceedings of the National Academy of Sciences*, 108(9):3465–3472. doi:https://dx.doi.org/10.1073/pnas.1100480108.

Lawrence, D. M., Thornton, P. E., Oleson, K. W., and Bonan, G. B. (2007). The Partitioning of Evapotranspiration into Transpiration, Soil Evaporation, and Canopy Evaporation in a GCM: Impacts on Land-Atmosphere Interaction. *Journal of Hydrometeorology*, 8:862–880. doi:https://dx.doi.org/10.1175/JHM596.1.

Leavesley, G., Lichty, R., Troutman, B., and Saindon, L. (1983). *Precipitation Runoff Modeling System–User's Manual*. United States Geological Survey, Denver, Colorado. Water Resources Investigations Report 83–4238.

Leavesley, G. H., Markstrom, S. L., and Viger, R. J. (2006). USGS modular modeling system (MMS)-precipitation-runoff modeling system (PRMS). In Singh, V. P. and Frevert, D. K., editors, *Watershed models*, pages 159–177. CRC Press Boca Raton, FL.

Lee, E., Chase, T. N., Rajagopalan, B., Barry, R. G., Biggs, T. W., and Lawrence, P. J. (2009). Effects of irrigation and vegetation activity on early Indian summer monsoon variability. *International Journal of Climatology*, 29(4):573–581. doi:https://dx.doi.org/10.1002/joc.1721.

Lewallen, R. (2005). 4 major principles of object-oriented programming. 19-july. Available from: http://codebetter.com/raymondlewallen/2005/07/19/4-major-principles-of-object-oriented-programming/ [Accessed 11-June-2016].

Li, C. and Yanai, M. (1996). The onset and interannual variability of the Asian summer monsoon in relation to land-sea thermal contrast. *Journal of Climate*, 9(2):358–375.

Liang, X., Lettenmaier, D. P., Wood, E. F., and Burges, S. J. (1994). A simple hydrologically based model of land surface water and energy fluxes for general circulation models. *Journal of Geophysical Research: Atmospheres*, 99(D7):14415–14428. doi:https://dx.doi.org/10.1029/94JD00483.

Liaw, A. and Wiener, M. (2002). Classification and regression by randomForest. *R news*, 2(3):18–22.

Liu, Y., Gupta, H., Springer, E., and Wagener, T. (2008). Linking science with environmental decision making: Experiences from an integrated modeling approach to supporting sustainable water resources management. *Environmental Modelling & Software*, 23(7):846–858. doi:https://dx.doi.org/10.1016/j.envsoft.2007.10.007.

Lloyd, W., David, O., Ascough, J., Rojas, K., Carlson, J., Leavesley, G., Krause, P., Green, T., and Ahuja, L. (2011). Environmental modeling framework invasiveness: Analysis and implications. *Environmental Modelling & Software*, 26(10):1240–1250. doi:https://dx.doi.org/10.1016/j.envsoft.2011.03.011.

Lobell, D., Bala, G., Mirin, A., Phillips, T., Maxwell, R., and Rotman, D. (2009). Regional differences in the influence of irrigation on climate. *Journal of Climate*, 22(8):2248–2255. doi:https://dx.doi.org/10.1175/2008JCLI2703.1.

Lobell, D. B., Bala, G., and Duffy, P. B. (2006). Biogeophysical impacts of cropland management changes on climate. *Geophysical Research Letters*, 33(6). doi:https://dx.doi.org/10.1029/2005GL025492.

Lobell, D. B., Bonfils, C., and Faurès, J. (2008). The role of irrigation expansion in past and future temperature trends. *Earth Interactions*, 12(3):1–11. doi:https://dx.doi.org/10.1175/2007EI241.1.

Long, D., Chen, X., Scanlon, B. R., Wada, Y., Hong, Y., Singh, V. P., Chen, Y., Wang, C., Han, Z., and Yang, W. (2016). Have GRACE satellites overestimated groundwater depletion in the Northwest India Aquifer? *Scientific Reports*, 6:24398.

Loucks, D. P. (2015). Debates-Perspectives on socio-hydrology: Simulating hydrologic-human interactions. *Water Resources Research*, 51(6):4789–4794. doi:https://dx.doi.org/10.1002/2015WR017002.

Loveland, T. R., Reed, B. C., Brown, J. F., Ohlen, D. O., Zhu, Z., Yang, L., and Merchant, J. W. (2000). Development of a global land cover characteristics database and IGBP DISCover from 1 km AVHRR data. *International Journal of Remote Sensing*, 21(6-7):1303–1330. doi:https://dx.doi.org/10.1080/014311600210191.

Ludwig, M. (2011). *Modelling and Architecture of a Generic Framework for Integrative Environmental Simulations*. PhD thesis, Ludwig-Maximilians-Universität München.

MacDonald, A. M., Bonsor, H. C., Ahmed, K. M., Burgess, W. G., Basharat, M., Calow, R. C., Dixit, A., Foster, S. S. D., Gopal, K., Lapworth, D. J., Lark, R. M., Moench, M., Mukherjee, A., Rao, M. S., Shamsudduha, M., Smith, L., Taylor, R. G., Tucker, J., van Steenbergen, F., and Yadav, S. K. (2016). Groundwater quality and depletion in the Indo-Gangetic Basin mapped from in situ observations. *Nature Geoscience*, 9(10):762–766.

Magliocca, N. R. and Ellis, E. C. (2013). Using pattern-oriented modeling (POM) to cope with uncertainty in multi-scale agent-based models of land change: POM in multi-scale ABMs of land change. *Transactions in GIS*, 17(6):883–900. doi:https://dx.doi.org/10.1111/tgis.12012.

Mallya, G., Mishra, V., Niyogi, D., Tripathi, S., and Govindaraju, R. S. (2016). Trends and variability of droughts over the Indian monsoon region. *Weather and Climate Extremes*. doi:https://dx.doi.org/10.1016/j.wace.2016.01.002.

Mas, J., Kolb, M., Paegelow, M., Camacho Olmedo, M. T., and Houet, T. (2014). Inductive pattern-based land use/cover change models: A comparison of four software packages. *Environmental Modelling & Software*, 51:94–111. doi:https://dx.doi.org/10.1016/j.envsoft.2013.09.010.

Mascaro, J., Asner, G. P., Knapp, D. E., Kennedy-Bowdoin, T., Martin, R. E., Anderson, C., Higgins, M., and Chadwick, K. D. (2014). A tale of two "Forests": Random Forest machine learning aids tropical forest carbon mapping. *PLoS ONE*, 9(1):e85993. doi:https://dx.doi.org/10.1371/journal.pone.0085993.

MassGIS (2015). Massachusetts geographic information system, massgis.

Maxwell, R. M. and Miller, N. L. (2005). Development of a coupled land surface and groundwater model. *Journal of Hydrometeorology*, 6(3):233–247.

McDonnell, J. J., Sivapalan, M., Vaché, K., Dunn, S., Grant, G., Haggerty, R., Hinz, C., Hooper, R., Kirchner, J., Roderick, M. L., Selker, J., and Weiler, M. (2007). Moving beyond heterogeneity and process complexity: A new vision for watershed hydrology. *Water Resources Research*, 43(7):n/a–n/a. doi:https://dx.doi.org/10.1029/2006WR005467.

McGuire, K., Weiler, M., and McDonnell, J. (2007). Integrating tracer experiments with modeling to assess runoff processes and water transit times. *Advances in Water Resources*, 30(4):824–837. doi:https://dx.doi.org/10.1016/j.advwatres.2006.07.004.

McMillan, H., Montanari, A., Cudennec, C., Savenjie, H., Kreibich, H., Krüger, T., Liu, J., Meija, A., van Loon, A., Aksoy, H., Di Baldassarre, G., Huang, Y., Mazvimavi, D., Rogger, M., Bellie, S., Bibikova, T., Castellarin, A., Chen, Y., Finger, D., Gelfan, A., Hannah, D., Hoekstra, A., Li, H., Maskey, S., Mathevet, T., Mijic, A., Acuña, A. P., Polo, M. J., Rosales, V., Smith, P., Viglione, A., Srinivasan, V., Toth, E., van Nooyen, R., and Xia, J. (2016). Panta Rhei 2013-2015: Global perspectives on hydrology, society and change. *Hydrological Sciences Journal*. doi:https://dx.doi.org/10.1080/02626667.2016.1159308.

McPhillips, T., Bowers, S., Zinn, D., and Ludäscher, B. (2009). Scientific workflow design for mere mortals. *Future Generation Computer Systems*, 25(5):541–551. doi:https://dx.doi.org/10.1016/j.future.2008.06.013.

Meehl, G. (1994). Influence of the land surface in the Asian summer monsoon: External conditions versus internal feedbacks. *Journal of Climate*, 7:1033–1049.

Meiyappan, P., Dalton, M., O'Neill, B. C., and Jain, A. K. (2014). Spatial modeling of agricultural land use change at global scale. *Ecological Modelling*, 291:152–174. doi:https://dx.doi.org/10.1016/j.ecolmodel.2014.07.027.

Meiyappan, P. and Jain, A. K. (2012). Three distinct global estimates of historical land-cover change and land-use conversions for over 200 years. *Frontiers of Earth Science*, 6(2):122–139. doi:https://dx.doi.org/10.1007/s11707-012-0314-2.

Montanari, A. and Koutsoyiannis, D. (2012). A blueprint for process-based modeling of uncertain hydrological systems. *Water Resources Research*, 48(9):n/a–n/a. doi:https://dx.doi.org/10.1029/2011WR011412.

Montanari, A., Young, G., Savenije, H., Hughes, D., Wagener, T., Ren, L., Koutsoyiannis, D., Cudennec, C., Toth, E., Grimaldi, S., Blöschl, G., Sivapalan, M., Beven, K., Gupta, H., Hipsey, M., Schaefli, B., Arheimer, B., Boegh, E., Schymanski, S., Di Baldassarre, G., Yu, B., Hubert, P., Huang, Y., Schumann, A., Post, D., Srinivasan, V., Harman, C., Thompson, S., Rogger, M., Viglione, A., McMillan, H., Characklis, G., Pang, Z., and Belyaev, V. (2013). "Panta Rhei—Everything flows": Change in hydrology and society—The IAHS scientific decade 2013–2022. *Hydrological Sciences Journal*, 58(6):1256–1275. doi:https://dx.doi.org/10.1080/02626667.2013.809088.

Moore, R. V. and Tindall, C. I. (2005). An overview of the open modelling interface and environment (the OpenMI). *Environmental Science & Policy*, 8(3):279–286. doi:https://dx.doi.org/10.1016/j.envsci.2005.03.009.

Moreira, E., Costa, S., Aguiar, A. P., Câmara, G., and Carneiro, T. (2009). Dynamical coupling of multiscale land change models. *Landscape Ecology*, 24(9):1183–1194. doi:https://dx.doi.org/10.1007/s10980-009-9397-x.

Morin, A., Urban, J., Adams, P. D., Foster, I., Sali, A., Baker, D., and Sliz, P. (2012). Shining light into black boxes. *Science*, 336(6078):159–160.

Moulds, S., Buytaert, W., and Mijic, A. (2015). An open and extensible framework for spatially explicit land use change modelling: the lulcc R package. *Geoscientific Model Development*, 8(10):3215–3229. doi:https://dx.doi.org/10.5194/gmd-8-3215-2015.

Mount, N., Maier, H., Toth, E., Elshorbagy, A., Solomatine, D., Chang, F., and Abrahart, R. (2016). Data-driven modelling approaches for socio-hydrology: Opportunities and challenges within the Panta Rhei Science Plan. *Hydrological Sciences Journal*. doi:https://dx.doi.org/10.1080/02626667.2016.1159683.

Mulia, R., Widayati, A., Suyanto, Agung, P., and Zulkarnain, M. T. (2014). Low carbon emission development strategies for Jambi, Indonesia: simulation and trade-off analysis

using the FALLOW model. *Mitigation and Adaptation Strategies for Global Change*, 19(6):773–788. doi:https://dx.doi.org/10.1007/s11027-013-9485-8.

National Remote Sensing Agency (2007). Natural resources census: national land use and land cover mapping using multi-temporal AWiFS data, project report. Publication No. NRSA/LULC/1:250K/2007-1.

Nativi, S., Mazzetti, P., and Geller, G. N. (2013). Environmental model access and interoperability: The GEO model web initiative. *Environmental Modelling & Software*, 39:214–228. doi:https://dx.doi.org/10.1016/j.envsoft.2012.03.007.

Naylor, R. (2011). Expanding the boundaries of agricultural development. *Food Security*, 3(2):233–251. doi:https://dx.doi.org/10.1007/s12571-011-0123-6.

Nazemi, A. and Wheater, H. S. (2015a). On inclusion of water resource management in Earth system models—Part 1: Problem definition and representation of water demand. *Hydrology and Earth System Sciences*, 19(1):33–61. doi:https://dx.doi.org/10.5194/hess-19-33-2015.

Nazemi, A. and Wheater, H. S. (2015b). On inclusion of water resource management in Earth system models—Part 2: Representation of water supply and allocation and opportunities for improved modeling. *Hydrology and Earth System Sciences*, 19(1):63–90. doi:https://dx.doi.org/10.5194/hess-19-63-2015.

Neumann, K., Stehfest, E., Verburg, P. H., Siebert, S., Müller, C., and Veldkamp, T. (2011). Exploring global irrigation patterns: A multilevel modelling approach. *Agricultural Systems*, 104(9):703–713. doi:https://dx.doi.org/10.1016/j.agsy.2011.08.004.

Niranjan Kumar, K., Rajeevan, M., Pai, D., Srivastava, A., and Preethi, B. (2013). On the observed variability of monsoon droughts over India. *Weather and Climate Extremes*, 1:42–50. doi:https://dx.doi.org/10.1016/j.wace.2013.07.006.

Niyogi, D., Kishtawal, C., Tripathi, S., and Govindaraju, R. S. (2010). Observational evidence that agricultural intensification and land use change may be reducing the Indian summer monsoon rainfall. *Water Resources Research*, 46(3):n/a–n/a. doi:https://dx.doi.org/10.1029/2008WR007082.

O'Keeffe, J., Buytaert, W., Mijic, A., Brozovic, N., and Sinha, R. (2015). The use of semi-structured interviews for the characterisation of farmer irrigation practices. *Hydrology and Earth System Sciences Discussions*, 12(8):8221–8246. doi:https://dx.doi.org/10.5194/hessd-12-8221-2015.

O'Keeffe, J., Buytaert, W., Mijic, A., Brozović, N., and Sinha, R. (2016). The use of semi-structured interviews for the characterisation of farmer irrigation practices. *Hydrology and Earth System Sciences*, 20(5):1911–1924. doi:https://dx.doi.org/10.5194/hess-20-1911-2016.

Osborne, T., Gornall, J., Hooker, J., Williams, K., Wiltshire, A., Betts, R., and Wheeler, T. (2015). JULES-crop: a parametrisation of crops in the Joint UK Land Environment Simulator. *Geoscientific Model Development*, 8(4):1139–1155. doi:https://dx.doi.org/10.5194/gmd-8-1139-2015.

Ousterhout, J. K. (1998). Scripting: Higher level programming for the 21st century. *Computer*, 31(3):23–30.

Overmars, K., de Koning, G., and Veldkamp, A. (2003). Spatial autocorrelation in multi-scale land use models. *Ecological Modelling*, 164(2-3):257–270. doi:https://dx.doi.org/10.1016/S0304-3800(03)00070-X.

Overmars, K. P. and Verburg, P. H. (2006). Multilevel modelling of land use from field to village level in the Philippines. *Agricultural Systems*, 89(2-3):435–456. doi:https://dx.doi.org/10.1016/j.agsy.2005.10.006.

Overmars, K. P., Verburg, P. H., and Veldkamp, T. A. (2007). Comparison of a deductive and an inductive approach to specify land suitability in a spatially explicit land use model. *Land Use Policy*, 24(3):584–599. doi:https://dx.doi.org/10.1016/j.landusepol.2005.09.008.

Pai, D. S., Sridhar, L., Rajeevan, M., Sreejith, O. P., Satbhai, N. S., and Mukhopadhyay, B. (2014). Development of a new high spatial resolution (0.25$\times$ 0.25) long period (1901âĂŞ2010) daily gridded rainfall data set over India and its comparison with existing data sets over the region. *Mausam*, 65(1):1–18.

Panagos, P., Jones, A., Bosco, C., and Kumar, P. S. (2011). European digital archive on soil maps (EuDASM): preserving important soil data for public free access. *International Journal of Digital Earth*, 4(5):434–443. doi:https://dx.doi.org/10.1080/17538947.2011.596580.

Panda, D. K. and Wahr, J. (2016). Spatiotemporal evolution of water storage changes in India from the updated GRACE-derived gravity records. *Water Resources Research*, 52(1):135–149. doi:https://dx.doi.org/10.1002/2015WR017797.

Parker, D. J., Willetts, P., Birch, C., Turner, A. G., Marsham, J. H., Taylor, C. M., Kolusu, S., and Martin, G. M. (2016). The interaction of moist convection and mid-level dry air in the advance of the onset of the Indian monsoon: Moist convection and mid-level

dry air in Indian monsoon onset. *Quarterly Journal of the Royal Meteorological Society*. doi:https://dx.doi.org/10.1002/qj.2815.

Pebesma, E. (2012). spacetime: Spatio-temporal data in R. *Journal of Statistical Software*, 51(7):1–30.

Pebesma, E. J. (2004). Multivariable geostatistics in S: the gstat package. *Computers & Geosciences*, 30(7):683–691. doi:https://dx.doi.org/10.1016/j.cageo.2004.03.012.

Pebesma, E. J. and Bivand, R. S. (2005). Classes and methods for spatial data in R. *R News*, 5(2):9–13.

Pechlivanidis, I. G., Jackson, B. M., Mcintyre, N. R., and Wheater, H. S. (2011). Catchment scale hydrological modelling: a review of model types, calibration approaches and uncertainty analysis methods in the context of recent developments in technology and applications. *Global NEST journal*, 13(3):193–214.

Peckham, S. D. and Goodall, J. L. (2013). Driving plug-and-play models with data from web services: A demonstration of interoperability between CSDMS and CUAHSI-HIS. *Computers & Geosciences*, 53:154–161. doi:https://dx.doi.org/10.1016/j.cageo.2012.04.019.

Peng, R. D. (2011). Reproducible research in computational science. *Science*, 334(6060):1226–1227. doi:https://dx.doi.org/10.1126/science.1213847.

Pérez-Vega, A., Mas, J., and Ligmann-Zielinska, A. (2012). Comparing two approaches to land use/cover change modeling and their implications for the assessment of biodiversity loss in a deciduous tropical forest. *Environmental Modelling & Software*, 29(1):11–23. doi:https://dx.doi.org/10.1016/j.envsoft.2011.09.011.

Petzoldt, T. and Rinke, K. (2007). Simecol: an object-oriented framework for ecological modeling in R. *Journal of Statistical Software*, 22(9):1–31.

Pielke, R. A., Marland, G., Betts, R. A., Chase, T. N., Eastman, J. L., Niles, J. O., Niyogi, D. D. S., and Running, S. W. (2002). The influence of land-use change and landscape dynamics on the climate system: relevance to climate-change policy beyond the radiative effect of greenhouse gases. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 360:1705–19. doi:https://dx.doi.org/10.1098/rsta.2002.1027.

Pielke, R. A., Pitman, A., Niyogi, D., Mahmood, R., McAlpine, C., Hossain, F., Goldewijk, K. K., Nair, U., Betts, R., Fall, S., Reichstein, M., Kabat, P., and de Noblet, N. (2011). Land use/land cover changes and climate: modeling analysis and observational evidence. *Wiley Interdisciplinary Reviews: Climate Change*, 2(6):828–850. doi:https://dx.doi.org/10.1002/wcc.144.

Pingali, P. L. (2012). Green Revolution: Impacts, limits, and the path ahead. *Proceedings of the National Academy of Sciences*, 109(31):12302–12308. doi:https://dx.doi.org/10.1073/pnas.0912953109.

Pinzon, J. and Tucker, C. (2014). A Non-Stationary 1981–2012 AVHRR NDVI3g time series. *Remote Sensing*, 6(8):6929–6960. doi:https://dx.doi.org/10.3390/rs6086929.

Pitman, A. J. (2003). The evolution of, and revolution in, land surface schemes designed for climate models. *International Journal of Climatology*, 23(5):479–510. doi:https://dx.doi.org/10.1002/joc.893.

Pitman, A. J., de Noblet-Ducoudré, N., Cruz, F. T., Davin, E. L., Bonan, G. B., Brovkin, V., Claussen, M., Delire, C., Ganzeveld, L., Gayler, V., van den Hurk, B. J. J. M., Lawrence, P. J., van der Molen, M. K., Müller, C., Reick, C. H., Seneviratne, S. I., Strengers, B. J., and Voldoire, A. (2009). Uncertainties in climate responses to past land cover change: First results from the LUCID intercomparison study. *Geophysical Research Letters*, 36(14). doi:https://dx.doi.org/10.1029/2009GL039076.

Pokhrel, Y., Hanasaki, N., Koirala, S., Cho, J., Yeh, P. J., Kim, H., Kanae, S., and Oki, T. (2012). Incorporating anthropogenic water regulation modules into a land surface model. *Journal of Hydrometeorology*, 13(1):255–269. doi:https://dx.doi.org/10.1175/JHM-D-11-013.1.

Pontius, R. G., Boersma, W., Castella, J., Clarke, K., Nijs, T., Dietzel, C., Duan, Z., Fotsing, E., Goldstein, N., Kok, K., Koomen, E., Lippitt, C. D., McConnell, W., Mohd Sood, A., Pijanowski, B., Pithadia, S., Sweeney, S., Trung, T. N., Veldkamp, A. T., and Verburg, P. H. (2007). Comparing the input, output, and validation maps for several models of land change. *The Annals of Regional Science*, 42(1):11–37. doi:https://dx.doi.org/10.1007/s00168-007-0138-2.

Pontius, R. G., Huffaker, D., and Denman, K. (2004a). Useful techniques of validation for spatially explicit land-change models. *Ecological Modelling*, 179(4):445–461. doi:https://dx.doi.org/10.1016/j.ecolmodel.2004.05.010.

Pontius, R. G. and Li, X. (2010). Land transition estimates from erroneous maps. *Journal of Land Use Science*, 5(1):31–44. doi:https://dx.doi.org/10.1080/17474230903222473.

Pontius, R. G. and Malanson, J. (2005). Comparison of the structure and accuracy of two land change models. *International Journal of Geographical Information Science*, 19(2):243–265. doi:https://dx.doi.org/10.1080/13658810410001713434.

Pontius, R. G. and Parmentier, B. (2014). Recommendations for using the relative operating characteristic (ROC). *Landscape Ecology*. doi:https://dx.doi.org/10.1007/s10980-013-9984-8.

Pontius, R. G., Peethambaram, S., and Castella, J. (2011). Comparison of three maps at multiple resolutions: A case study of land change simulation in Cho Don District, Vietnam. *Annals of the Association of American Geographers*, 101(1):45–62. doi:https://dx.doi.org/10.1080/00045608.2010.517742.

Pontius, R. G. and Schneider, L. C. (2001). Land-cover change model validation by an ROC method for the Ipswich watershed, Massachusetts, USA. *Agriculture, Ecosystems & Environment*, 85(1):239–248.

Pontius, R. G., Shusas, E., and McEachern, M. (2004b). Detecting important categorical land changes while accounting for persistence. *Agriculture, Ecosystems & Environment*, 101(2-3):251–268. doi:https://dx.doi.org/10.1016/j.agee.2003.09.008.

Pontius, R. G. and Spencer, J. (2005). Uncertainty in extrapolations of predictive land-change models. *Environment and Planning B: Planning and Design*, 32(2):211–230. doi:https://dx.doi.org/10.1068/b31152.

Portmann, F. T., Siebert, S., and Döll, P. (2010). MIRCA2000-Global monthly irrigated and rainfed crop areas around the year 2000: A new high-resolution data set for agricultural and hydrological modeling. *Global Biogeochemical Cycles*, 24(1):n/a–n/a. doi:https://dx.doi.org/10.1029/2008GB003435.

Prasanna, V. (2014). Impact of monsoon rainfall on the total foodgrain yield over India. *Journal of Earth System Science*, 123(5):1129–1145.

Puma, M. J. and Cook, B. I. (2010). Effects of irrigation on global climate during the 20th century. *Journal of Geophysical Research*, 115(D16). doi:https://dx.doi.org/10.1029/2010JD014122.

R Core Team (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rajeevan, M., Bhate, J., and Jaswal, A. K. (2008). Analysis of variability and trends of extreme rainfall events over India using 104 years of gridded daily rainfall data. *Geophysical Research Letters*, 35(18). doi:https://dx.doi.org/10.1029/2008GL035143.

Rajeevan, M., Bhate, J., Kale, J. D., and Lal, B. (2006). High resolution daily gridded rainfall data for the Indian region: Analysis of break and active monsoon spells. *Current Science*, 91(3):296–306.

Ramanathan, V., Chung, C., Kim, D., Bettge, T., Buja, L., Kiehl, J. T., Washington, W. M., Fu, Q., Sikka, D. R., and Wild, M. (2005). Atmospheric brown clouds: Impacts on South Asian climate and hydrological cycle. *Proceedings of the National Academy of Sciences of the United States of America*, 102(15):5326–5333.

Ramankutty, N. and Foley, J. A. (1999). Estimating historical changes in global land cover: Croplands from 1700 to 1992. *Global Biogeochemical Cycles*, 13(4):997–1027. doi:https://dx.doi.org/10.1029/1999GB900046.

Ravindranath, N., Sita Lakshmi, C., Manuvie, R., and Balachandra, P. (2011). Biofuel production and implications for land use, food production and environment in India. *Energy Policy*, 39(10):5737–5745. doi:https://dx.doi.org/10.1016/j.enpol.2010.07.044.

Ray, D. K. and Pijanowski, B. C. (2010). A backcast land use change model to generate past land use maps: application and validation at the Muskegon River watershed of Michigan, USA. *Journal of Land Use Science*, 5(1):1–29. doi:https://dx.doi.org/10.1080/17474230903150799.

Reed, S., Koren, V., Smith, M., Zhang, Z., Moreda, F., Seo, D., and DMIP Participants, a. (2004). Overall distributed model intercomparison project results. *Journal of Hydrology*, 298(1-4):27–60. doi:https://dx.doi.org/10.1016/j.jhydrol.2004.03.031.

Reusser, D. (2016). *RObsDat: Data Management for Hydrology and Beyond Using the Observations Data Model*. R package version 16.03.

Richards, L. A. (1931). Capillary Conduction of Liquids Through Porous Mediums. *Physics*, 1:318–333. doi:https://dx.doi.org/10.1063/1.1745010.

Rigon, R., Bertoldi, G., and Over, T. M. (2006). GEOtop: a distributed hydrological model with coupled water and energy budgets. *Journal of Hydrometeorology*, 7(3):371–388.

Ripley, B. D. and Hornik, K. (2001). Date-time classes. *R News*, 1(2):8–11.

Rodell, M., Velicogna, I., and Famiglietti, J. S. (2009). Satellite-based estimates of groundwater depletion in India. *Nature*, 460(7258):999–1002. doi:https://dx.doi.org/10.1038/nature08238.

Rosa, I. M. D., Ahmed, S. E., and Ewers, R. M. (2014). The transparency, reliability and utility of tropical rainforest land-use and land-cover change models. *Global Change Biology*, 20(6):1707–1722. doi:https://dx.doi.org/10.1111/gcb.12523.

Rosa, I. M. D., Purves, D., Souza, C., and Ewers, R. M. (2013). Predictive modelling of contagious deforestation in the Brazilian Amazon. *PLoS ONE*, 8(10):e77231. doi:https://dx.doi.org/10.1371/journal.pone.0077231.

Rost, S., Gerten, D., Bondeau, A., Lucht, W., Rohwer, J., and Schaphoff, S. (2008). Agricultural green and blue water consumption and its influence on the global water system. *Water Resources Research*, 44(9):n/a–n/a. doi:https://dx.doi.org/10.1029/2007WR006331.

Sacks, W. J., Cook, B. I., Buenning, N., Levis, S., and Helkowski, J. H. (2009). Effects of global irrigation on the near-surface climate. *Climate Dynamics*, 33(2-3):159–175. doi:https://dx.doi.org/10.1007/s00382-008-0445-z.

Saeed, F., Hagemann, S., and Jacob, D. (2009). Impact of irrigation on the South Asian summer monsoon. *Geophysical Research Letters*, 36(20). doi:https://dx.doi.org/10.1029/2009GL040625.

Saha, A., Ghosh, S., Sahana, A. S., and Rao, E. P. (2014). Failure of CMIP5 climate models in simulating post-1950 decreasing trend of Indian monsoon: CMIP5 models and Indian monsoon. *Geophysical Research Letters*, 41(20):7323–7330. doi:https://dx.doi.org/10.1002/2014GL061573.

Saha, S. K., Halder, S., Kumar, K. K., and Goswami, B. N. (2011). Pre-onset land surface processes and 'internal' interannual variabilities of the Indian summer monsoon. *Climate Dynamics*, 36(11-12):2077–2089. doi:https://dx.doi.org/10.1007/s00382-010-0886-z.

Salmon, J., Friedl, M. A., Frolking, S., Wisser, D., and Douglas, E. M. (2015). Global rain-fed, irrigated, and paddy croplands: A new high resolution map derived from remote sensing, crop inventories and climate data. *International Journal of Applied Earth Observation and Geoinformation*, 38:321–334. doi:https://dx.doi.org/10.1016/j.jag.2015.01.014.

Sarkar, A., Patil, S., Hugar, L. B., and vanLoon, G. (2011). Sustainability of current agriculture practices, community perception, and implications for ecosystem health: An Indian study. *EcoHealth*, 8(4):418–431. doi:https://dx.doi.org/10.1007/s10393-011-0723-9.

Savenije, H. H. G. (2009). HESS opinions "The art of hydrology". *Hydrology and Earth System Sciences*, 13(2):157–161.

Savenije, H. H. G., Hoekstra, A. Y., and van der Zaag, P. (2014). Evolving water science in the Anthropocene. *Hydrology and Earth System Sciences*, 18(1):319–332. doi:https://dx.doi.org/10.5194/hess-18-319-2014.

Schade, S., Ostländer, N., Granell, C., Schulz, M., McInerney, D., Dubois, G., Vaccari, L., Chinosi, M., Díaz, L., Bastin, L., et al. (2012). Which service interfaces fit the

model web? In *GEOProcessing 2012: The Fourth International Conference on Advanced Geographic Information Systems, Applications, and Services, Wilmington (US)*, pages 1–6.

Schaldach, R., Alcamo, J., Koch, J., Kölking, C., Lapola, D. M., Schüngel, J., and Priess, J. A. (2011). An integrated approach to modelling land-use change on continental and global scales. *Environmental Modelling & Software*, 26(8):1041–1051. doi:https://dx.doi.org/10.1016/j.envsoft.2011.02.013.

Schmitz, O., Karssenberg, D., van Deursen, W., and Wesseling, C. (2009). Linking external components to a spatio-temporal modelling framework: Coupling MODFLOW and PCRaster. *Environmental Modelling & Software*, 24(9):1088–1099. doi:https://dx.doi.org/10.1016/j.envsoft.2009.02.018.

Schneider, L. C., Pontius Jr, G., et al. (2001). Modeling land-use change in the Ipswich watershed, Massachusetts, USA. *Agriculture, Ecosystems & Environment*, 85(1):83–94.

Scott, C. A. and Sharma, B. (2009). Energy supply and the expansion of groundwater irrigation in the Indus-Ganges basin. *International Journal of River Basin Management*, 7(2):119–124.

Sen Roy, S., Mahmood, R., Niyogi, D., Lei, M., Foster, S. A., Hubbard, K. G., Douglas, E., and Pielke, R. (2007). Impacts of the agricultural Green Revolution–induced land use changes on air temperatures in India. *Journal of Geophysical Research*, 112(D21). doi:https://dx.doi.org/10.1029/2007JD008834.

Sen Roy, S., Mahmood, R., Quintanar, A. I., and Gonzalez, A. (2011). Impacts of irrigation on dry season precipitation in India. *Theoretical and Applied Climatology*, 104(1-2):193–207. doi:https://dx.doi.org/10.1007/s00704-010-0338-z.

Seneviratne, S. I., Corti, T., Davin, E. L., Hirschi, M., Jaeger, E. B., Lehner, I., Orlowsky, B., and Teuling, A. J. (2010). Investigating soil moisture–climate interactions in a changing climate: A review. *Earth-Science Reviews*, 99(3-4):125–161. doi:https://dx.doi.org/10.1016/j.earscirev.2010.02.004.

Shah, T. (2009). Climate change and groundwater: India's opportunities for mitigation and adaptation. *Environmental Research Letters*, 4(3):035005. doi:https://dx.doi.org/10.1088/1748-9326/4/3/035005.

Shah, T., Singh, O. P., and Mukherji, A. (2006). Some aspects of South Asia's groundwater irrigation economy: analyses from a survey in India, Pakistan, Nepal Terai and Bangladesh. *Hydrogeology Journal*, 14(3):286–309. doi:https://dx.doi.org/10.1007/s10040-005-0004-1.

Shankar, P. V., Kulkarni, H., and Krishnan, S. (2011). India's groundwater challenge and the way forward. *Economic & Political Weekly*, 46(2):37.

Shao, Q., Sun, P., and Chen, Y. (2009). Efficiently discovering critical workflows in scientific explorations. *Future Generation Computer Systems*, 25(5):577–585. doi:https://dx.doi.org/10.1016/j.future.2008.06.005.

Sharmila, S., Joseph, S., Sahai, A., Abhilash, S., and Chattopadhyay, R. (2015). Future projection of Indian summer monsoon variability under climate change scenario: An assessment from CMIP5 climate models. *Global and Planetary Change*, 124:62–78. doi:https://dx.doi.org/10.1016/j.gloplacha.2014.11.004.

Shaw, E. (1994). *Hydrology in Practice*. Chapman & Hall, London, third edition.

Sheffield, J., Goteti, G., and Wood, E. F. (2006). Development of a 50-year high-resolution global dataset of meteorological forcings for land surface modeling. *Journal of Climate*, 19(13):3088–3111.

Shrestha, P., Sulis, M., Masbou, M., Kollet, S., and Simmer, C. (2014). A Scale-Consistent terrestrial systems modeling platform based on COSMO, CLM, and ParFlow. *Monthly Weather Review*, 142(9):3466–3483. doi:https://dx.doi.org/10.1175/MWR-D-14-00029.1.

Shukla, J. and Mintz, Y. (1982). Influence of land-surface evapotranspiration on the Earth's climate. *Science*, 215(4539):1498–1501.

Shukla, S. P., Puma, M. J., and Cook, B. I. (2014). The response of the South Asian summer monsoon circulation to intensified irrigation in global climate model simulations. *Climate Dynamics*, 42(1-2):21–36. doi:https://dx.doi.org/10.1007/s00382-013-1786-9.

Siebert, S., Döll, P., Hoogeveen, J., Faures, J., Frenken, K., and Feick, S. (2005). Development and validation of the global map of irrigation areas. *Hydrology and Earth System Sciences Discussions Discussions*, 2(4):1299–1327.

Siebert, S., Kummu, M., Porkka, M., Döll, P., Ramankutty, N., and Scanlon, B. R. (2015). A global data set of the extent of irrigated land from 1900 to 2005. *Hydrology and Earth System Sciences*, 19(3):1521–1545. doi:https://dx.doi.org/10.5194/hess-19-1521-2015.

Sing, T., Sander, O., Beerenwinkel, N., and Lengauer, T. (2005). ROCR: visualizing classifier performance in R. *Bioinformatics*, 21(20):3940–3941. doi:https://dx.doi.org/10.1093/bioinformatics/bti623.

Singh, D. (2016). South Asian monsoon: Tug of war on rainfall changes. *Nature Climate Change*, 6(1):20–22.

Singh, D., Tsiang, M., Rajaratnam, B., and Diffenbaugh, N. S. (2014). Observed changes in extreme wet and dry spells during the South Asian summer monsoon season. *Nature Climate Change*, 4(6):456–461. doi:https://dx.doi.org/10.1038/nclimate2208.

Singh, N. J., Kudrat, M., Jain, K., and Pandey, K. (2011). Cropping pattern of Uttar Pradesh using IRS-P6 (AWiFS) data. *International Journal of Remote Sensing*, 32(16):4511–4526. doi:https://dx.doi.org/10.1080/01431161.2010.489061.

Singh, R. B. (2000). Environmental consequences of agricultural development: a case study from the Green Revolution state of Haryana, India. *Agriculture, ecosystems & environment*, 82(1):97–103.

Sivakumar, B. (2008). Dominant processes concept, model simplification and classification framework in catchment hydrology. *Stochastic Environmental Research and Risk Assessment*, 22(6):737–748. doi:https://dx.doi.org/10.1007/s00477-007-0183-5.

Sivapalan, M. (2003). Prediction in ungauged basins: a grand challenge for theoretical hydrology. *Hydrological Processes*, 17(15):3163–3170. doi:https://dx.doi.org/10.1002/hyp.5155.

Sivapalan, M. (2009). The secret to 'doing better hydrological science': change the question! *Hydrological Processes*, 23(9):1391–1396. doi:https://dx.doi.org/10.1002/hyp.7242.

Sivapalan, M. and Blöschl, G. (2015). Time scale interactions and the coevolution of humans and water. *Water Resources Research*, 51(9):6988–7022. doi:https://dx.doi.org/10.1002/2015WR017896.

Sivapalan, M., Savenije, H. H. G., and Blöschl, G. (2012). Socio-hydrology: A new science of people and water. *Hydrological Processes*, 26(8):1270–1276. doi:https://dx.doi.org/10.1002/hyp.8426.

Smilovic, M., Gleeson, T., and Siebert, S. (2015). The limits of increasing food production with irrigation in India. *Food Security*, 7(4):835–856. doi:https://dx.doi.org/10.1007/s12571-015-0477-2.

Smith, M. B., Koren, V., Zhang, Z., Zhang, Y., Reed, S. M., Cui, Z., Moreda, F., Cosgrove, B. A., Mizukami, N., and Anderson, E. A. (2012). Results of the DMIP 2 Oklahoma experiments. *Journal of Hydrology*, 418-419:17–48. doi:https://dx.doi.org/10.1016/j.jhydrol.2011.08.056.

Soares-Filho, B. S., Coutinho Cerqueira, G., and Lopes Pennachin, C. (2002). DINAMICA—a stochastic cellular automata model designed to simulate the landscape dynamics in an Amazonian colonization frontier. *Ecological modelling*, 154(3):217–235.

Sohl, T., Reker, R., Bouchard, M., Sayler, K., Dornbierer, J., Wika, S., Quenzer, R., and Friesz, A. (2016). Modeled historical land use and land cover for the conterminous United States. *Journal of Land Use Science*, 11(4):476–499.

Sohl, T. L., Sayler, K. L., Drummond, M. A., and Loveland, T. R. (2007). The FORE-SCE model: a practical approach for projecting land cover change using scenario-based modeling. *Journal of Land Use Science*, 2(2):103–126. doi:https://dx.doi.org/10.1080/17474230701218202.

Sooraj, K. P., Terray, P., and Mujumdar, M. (2015). Global warming and the weakening of the Asian summer monsoon circulation: assessments from the CMIP5 models. *Climate Dynamics*, 45(1-2):233–252. doi:https://dx.doi.org/10.1007/s00382-014-2257-7.

Souty, F., Brunelle, T., Dumas, P., Dorin, B., Ciais, P., Crassous, R., Müller, C., and Bondeau, A. (2012). The Nexus Land-Use model version 1.0, an approach articulating biophysical potentials and economic dynamics to model competition for land-use. *Geoscientific Model Development*, 5(5):1297–1322. doi:https://dx.doi.org/10.5194/gmd-5-1297-2012.

Sperber, K. R., Annamalai, H., Kang, I., Kitoh, A., Moise, A., Turner, A., Wang, B., and Zhou, T. (2013). The Asian summer monsoon: an intercomparison of CMIP5 vs. CMIP3 simulations of the late 20th century. *Climate Dynamics*, 41(9-10):2711–2744. doi:https://dx.doi.org/10.1007/s00382-012-1607-6.

Srivastava, P., Singh, R., Tripathi, S., and Raghubanshi, A. S. (2016). An urgent need for sustainable thinking in agriculture – an Indian scenario. *Ecological Indicators*, 67:611–622. doi:https://dx.doi.org/10.1016/j.ecolind.2016.03.015.

Stehfast, E., van Vuuren, D., Kram, T., Bouwman, L., Alkemade, R., Bakkenes, M., Biemans, H., Bouwman, A., den Elzen, M., Janse, J., Lucas, P., van Minnen, J., Muller, M., and Prins, A. G. (2014). *Integrated Assessment of Global Environmental Change with IMAGE 3.0 - Model description and policy applications*. ISBN 978-94-91506-71-0.

Steiniger, S. and Hunter, A. J. (2013). The 2012 free and open source GIS software map – a guide to facilitate research, development, and adoption. *Computers, Environment and Urban Systems*, 39:136–150. doi:https://dx.doi.org/10.1016/j.compenvurbsys.2012.10.003.

Sterman, J. D. (2000). *Business dynamics: systems thinking and modeling for a complex world*, volume 19. Irwin/McGraw-Hill Boston.

Suhas, E., Neena, J. M., and Goswami, B. N. (2013). An Indian monsoon intraseasonal oscillations (MISO) index for real time monitoring and forecast verification. *Climate Dynamics*, 40(11-12):2605–2616. doi:https://dx.doi.org/10.1007/s00382-012-1462-5.

Sutton, R. T., Dong, B., and Gregory, J. M. (2007). Land/sea warming ratio in response to climate change: IPCC AR4 model results and comparison with observations. *Geophysical Research Letters*, 34(2). doi:https://dx.doi.org/10.1029/2006GL028164.

Svetnik, V., Liaw, A., Tong, C., Culberson, J., Sheridan, R., and Feuston, B. (2003). Random Forest: A classification and regression tool for compound classification and QSAR modeling. *Journal of Chemical Information and Modeling*, 43(6):1947–1958. doi:https://dx.doi.org/10.1021/ci034160g.

Syvitski, J., Paola, C., Slingerland, R., Furbish, D., Wiberg, P., and Tucker, G. (2004). Building a Community Surface Dynamics Modeling System: Rationale and strategy. a report from the scientific community to the national science foundation. Available from: http://www.nsf-margins.org/S2S/S2SWhitePaper.pdf [Accessed 28-March-2015].

Tayyebi, A., Pijanowski, B. C., Linderman, M., and Gratton, C. (2014). Comparing three global parametric and local non-parametric models to simulate land use change in diverse areas of the world. *Environmental Modelling & Software*, 59:202–221. doi:https://dx.doi.org/10.1016/j.envsoft.2014.05.022.

Tayyebi, A., Pijanowski, B. C., and Pekin, B. K. (2015). Land use legacies of the Ohio River Basin: Using a spatially explicit land use change model to assess past and future impacts on aquatic resources. *Applied Geography*, 57:100–111.

Thenkabail, P. S., Dheeravath, V., Biradar, C. M., Gangalakunta, O. R. P., Noojipady, P., Gurappa, C., Velpuri, M., Gumma, M., and Li, Y. (2009). Irrigated area maps and statistics of India using remote sensing and national statistics. *Remote Sensing*, 1(2):50–67. doi:https://dx.doi.org/10.3390/rs1020050.

Thenkabail, P. S., Schull, M., and Turral, H. (2005). Ganges and Indus river basin land use/land cover (LULC) and irrigated area mapping using continuous streams of MODIS data. *Remote Sensing of Environment*, 95(3):317–341. doi:https://dx.doi.org/10.1016/j.rse.2004.12.018.

Therneau, T., Atkinson, B., and Ripley, B. (2014). *rpart: Recursive Partitioning and Regression Trees*. R package version 4.1-8.

Tian, H., Banger, K., Bo, T., and Dadhwal, V. K. (2014). History of land use in India during 1880–2010: Large-scale land transformations reconstructed from satellite data and historical archives. *Global and Planetary Change*, 121:78–88. doi:https://dx.doi.org/10.1016/j.gloplacha.2014.07.005.

Trenberth, K. E., Dai, A., Rasmussen, R. M., and Parsons, D. B. (2003). The changing character of precipitation. *Bulletin of the American Meteorological Society*, 84(9):1205–1217. doi:https://dx.doi.org/10.1175/BAMS-84-9-1205.

Troy, T. J., Konar, M., Srinivasan, V., and Thompson, S. (2015). Moving sociohydrology forward: a synthesis across studies. *Hydrology and Earth System Sciences*, 19(8):3667–3679. doi:https://dx.doi.org/10.5194/hess-19-3667-2015.

Tsarouchi, G.-M. (2014). *Modelling land-use and climate change impacts on hydrology: the Upper Ganges river basin*. PhD thesis, Imperial College London.

Tsarouchi, G. M., Buytaert, W., and Mijic, A. (2014). Coupling a land-surface model with a crop growth model to improve ET flux estimations in the Upper Ganges basin, India. *Hydrology and Earth System Sciences*, 18(10):4223–4238. doi:https://dx.doi.org/10.5194/hess-18-4223-2014.

Tuinenburg, O. A., Hutjes, R. W. A., Stacke, T., Wiltshire, A., and Lucas-Picher, P. (2014). Effects of irrigation in India on the atmospheric water budget. *Journal of Hydrometeorology*, 15(3):1028–1050. doi:https://dx.doi.org/10.1175/JHM-D-13-078.1.

Turner, A. G. and Annamalai, H. (2012). Climate change and the South Asian summer monsoon. *Nature Climate Change*, 2(8):587–595. doi:https://dx.doi.org/10.1038/nclimate1495.

Turner, B. L., Lambin, E. F., and Reenberg, A. (2007). The emergence of land change science for global environmental change and sustainability. *Proceedings of the National Academy of Sciences*, 104(52):20666–20671.

Unnikrishnan, C. K., Rajeevan, M., and Vijaya Bhaskara Rao, S. (2015). A study on the role of land-atmosphere coupling on the South Asian monsoon climate variability using a regional climate model. *Theoretical and Applied Climatology*. doi:https://dx.doi.org/10.1007/s00704-015-1680-y.

Valcke, S., Balaji, V., Craig, A., DeLuca, C., Dunlap, R., Ford, R. W., Jacob, R., Larson, J., O'Kuinghttons, R., Riley, G. D., and Vertenstein, M. (2012). Coupling technologies for Earth System Modelling. *Geoscientific Model Development*, 5(6):1589–1596. doi:https://dx.doi.org/10.5194/gmd-5-1589-2012.

Van den Hoof, C., Hanert, E., and Vidale, P. L. (2011). Simulating dynamic crop growth with an adapted land surface model – JULES-SUCROS: model development and validation. *Agricultural and Forest Meteorology*, 151(2):137–153. doi:https://dx.doi.org/10.1016/j.agrformet.2010.09.011.

van Emmerik, T. H. M., Li, Z., Sivapalan, M., Pande, S., Kandasamy, J., Savenije, H. H. G., Chanan, A., and Vigneswaran, S. (2014). Socio-hydrologic modeling to understand and mediate the competition for water between agriculture development and environmental health: Murrumbidgee River basin, Australia. *Hydrology and Earth System Sciences*, 18(10):4239–4259. doi:https://dx.doi.org/10.5194/hess-18-4239-2014.

Van Genuchten, M. (1980). A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. *Soil Science Society of America Journal*, 44(5):892–898.

van Noordwijk, M. (2002). Scaling trade-offs between crop productivity, carbon stocks and biodiversity in shifting cultivation landscape mosaics: the FALLOW model. *Ecological Modelling*, 149(1):113–126.

van Vliet, J., Bregt, A. K., Brown, D. G., van Delden, H., Heckbert, S., and Verburg, P. H. (2016). A review of current calibration and validation practices in land-change modeling. *Environmental Modelling & Software*, 82:174–182. doi:https://dx.doi.org/10.1016/j.envsoft.2016.04.017.

van Vliet, J., Bregt, A. K., and Hagen-Zanker, A. (2011). Revisiting Kappa to account for change in the accuracy assessment of land-use change models. *Ecological Modelling*, 222(8):1367–1375. doi:https://dx.doi.org/10.1016/j.ecolmodel.2011.01.017.

Vanderlinden, K. and Giráldez, J. V. (2011). Field water capacity. In Gliński, J., Horabik, J., and Lipiec, J., editors, *Encyclopedia of Agrophysics*, pages 299–300. Springer Netherlands, Dordrecht. doi:https://dx.doi.org/10.1007/978-90-481-3585-1_58.

Varua, M., Ward, J., Maheshwari, B., Oza, S., Purohit, R., Hakimuddin, and Chinnasamy, P. (2016). Assisting community management of groundwater: Irrigator attitudes in two watersheds in Rajasthan and Gujarat, India. *Journal of Hydrology*, 537:171–186. doi:https://dx.doi.org/10.1016/j.jhydrol.2016.02.003.

Veldkamp, A. and Fresco, L. (1996a). CLUE: a conceptual model to study the conversion of land use and its effects. *Ecological modelling*, 85(2):253–270.

Veldkamp, A. and Fresco, L. (1996b). CLUE-CR: an integrated multi-scale model to simulate land use change scenarios in Costa Rica. *Ecological Modelling*, 91(1):231–248.

Veldkamp, A. and Lambin, E. F. (2001). Predicting land-use change. *Agriculture, ecosystems & environment*, 85(1):1–6.

Verburg, P. H., De Koning, G. H. J., Kok, K., Veldkamp, A., and Bouma, J. (1999). A spatial explicit allocation procedure for modelling the pattern of land use change based upon actual land use. *Ecological modelling*, 116(1):45–61.

Verburg, P. H., Neumann, K., and Nol, L. (2011). Challenges in using land use and land cover data for global change studies. *Global Change Biology*, 17(2):974–989. doi:https://dx.doi.org/10.1111/j.1365-2486.2010.02307.x.

Verburg, P. H. and Overmars, K. P. (2009). Combining top-down and bottom-up dynamics in land use modeling: exploring the future of abandoned farmlands in Europe with the Dyna-CLUE model. *Landscape Ecology*, 24(9):1167–1181. doi:https://dx.doi.org/10.1007/s10980-009-9355-7.

Verburg, P. H., Soepboer, W., Veldkamp, A., Limpiada, R., Espaldon, V., and Mastura, S. S. (2002). Modeling the spatial dynamics of regional land use: The CLUE-S model. *Environmental Management*, 30(3):391–405. doi:https://dx.doi.org/10.1007/s00267-002-2630-x.

Verburg, P. H., Tabeau, A., and Hatna, E. (2013). Assessing spatial uncertainties of land allocation using a scenario approach and sensitivity analysis: A study for land use in Europe. *Journal of Environmental Management*, 127:S132–S144. doi:https://dx.doi.org/10.1016/j.jenvman.2012.08.038.

Verburg, P. H. and Veldkamp, A. (2004). Projecting land use transitions at forest fringes in the Philippines at two spatial scales. *Landscape Ecology*, 19(1):77–98.

Vinnarasi, R. and Dhanya, C. T. (2016). Changing characteristics of extreme wet and dry spells of Indian monsoon rainfall: Changing characteristics of extremes. *Journal of Geophysical Research: Atmospheres*, 121(5):2146–2160. doi:https://dx.doi.org/10.1002/2015JD024310.

Voinov, A. and Shugart, H. H. (2013). 'Integronsters', integral and integrated modeling. *Environmental Modelling & Software*, 39:149–158. doi:https://dx.doi.org/10.1016/j.envsoft.2012.05.014.

Vörösmarty, C., Federer, C., and Schloss, A. (1998). Potential evaporation functions compared on US watersheds: Possible implications for global-scale water balance and terrestrial ecosystem modeling. *Journal of Hydrology*, 207(3):147 – 169. doi:https://dx.doi.org/http://dx.doi.org/10.1016/S0022-1694(98)00109-7.

Wada, Y., van Beek, L. P. H., and Bierkens, M. F. P. (2012). Nonsustainable groundwater sustaining irrigation: A global assessment. *Water Resources Research*, 48. doi:https://dx.doi.org/10.1029/2011WR010562.

Wada, Y., Wisser, D., Eisner, S., Flörke, M., Gerten, D., Haddeland, I., Hanasaki, N., Masaki, Y., Portmann, F. T., Stacke, T., Tessler, Z., and Schewe, J. (2013). Multimodel projections and uncertainties of irrigation water demand under climate change. *Geophysical Research Letters*, 40(17):4626–4632. doi:https://dx.doi.org/10.1002/grl.50686.

Wagener, T., Sivapalan, M., Troch, P., and Woods, R. (2007). Catchment classification and hydrologic similarity. *Geography Compass*, 1(4):901–931. doi:https://dx.doi.org/10.1111/j.1749-8198.2007.00039.x.

Wassenaar, T., Gerber, P., Verburg, P., Rosales, M., Ibrahim, M., and Steinfeld, H. (2007). Projecting land use changes in the neotropics: The geography of pasture expansion into forest. *Global Environmental Change*, 17(1):86–104. doi:https://dx.doi.org/10.1016/j.gloenvcha.2006.03.007.

Weiler, M. and Beven, K. (2015). Do we need a Community Hydrological Model? *Water Resources Research*, pages n/a–n/a. doi:https://dx.doi.org/10.1002/2014WR016731.

Wheater, H. S. (2002). Progress in and prospects for fluvial flood modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 360(1796):1409–1431. doi:https://dx.doi.org/10.1098/rsta.2002.1007.

Wheater, H. S., Jakeman, A. J., and Beven, K. J. (1993). Progress and directions in rainfall-runoff modelling. In Jakeman, A. J., Beck, M. B., and McAleer, M. J., editors, *Modelling Change in Environmental Systems*, pages 101–132. Wiley.

Wilson, G., Aruliah, D. A., Brown, C. T., Chue Hong, N. P., Davis, M., Guy, R. T., Haddock, S. H. D., Huff, K. D., Mitchell, I. M., Plumbley, M. D., Waugh, B., White, E. P., and Wilson, P. (2014). Best practices for scientific computing. *PLoS Biology*, 12(1):e1001745. doi:https://dx.doi.org/10.1371/journal.pbio.1001745.

Wisser, D., Fekete, B. M., Vörösmarty, C. J., and Schumann, A. H. (2010). Reconstructing 20th century global hydrography: a contribution to the Global Terrestrial Network-Hydrology (GTN-H). *Hydrology and Earth System Sciences*, 14(1):1–24.

Wisser, D., Frolking, S., Douglas, E. M., Fekete, B. M., Vörösmarty, C. J., and Schumann, A. H. (2008). Global irrigation water demand: Variability and uncertainties arising from agricultural and climate data sets. *Geophysical Research Letters*, 35(24). doi:https://dx.doi.org/10.1029/2008GL035296.

Wood, E. F., Lettenmaier, D. P., Liang, X., Lohmann, D., Boone, A., Chang, S., Chen, F., Dai, Y., Dickinson, R. E., Duan, Q., et al. (1998). The Project for Intercomparison of Land-surface Parameterization Schemes (PILPS) phase 2 (c) Red–Arkansas River basin experiment: 1. Experiment description and summary intercomparisons. *Global and Planetary Change*, 19(1):115–135.

Wood, E. F., Roundy, J. K., Troy, T. J., van Beek, L. P. H., Bierkens, M. F. P., Blyth, E., de Roo, A., Döll, P., Ek, M., Famiglietti, J., Gochis, D., van de Giesen, N., Houser, P., Jaffé, P. R., Kollet, S., Lehner, B., Lettenmaier, D. P., Peters-Lidard, C., Sivapalan, M.,

Sheffield, J., Wade, A., and Whitehead, P. (2011). Hyperresolution global land surface modeling: Meeting a grand challenge for monitoring Earth's terrestrial water. *Water Resources Research*, 47(5). doi:https://dx.doi.org/10.1029/2010WR010090.

Wood, E. F., Roundy, J. K., Troy, T. J., van Beek, R., Bierkens, M., Blyth, E., de Roo, A., Döll, P., Ek, M., Famiglietti, J., Gochis, D., van de Giesen, N., Houser, P., Jaffe, P., Kollet, S., Lehner, B., Lettenmaier, D. P., Peters-Lidard, C. D., Sivapalan, M., Sheffield, J., Wade, A. J., and Whitehead, P. (2012). Reply to comment by Keith J. Beven and Hannah L. Cloke on "Hyperresolution global land surface modeling: Meeting a grand challenge for monitoring Earth's terrestrial water". *Water Resources Research*, 48(1). doi:https://dx.doi.org/10.1029/2011WR011202.

Yang, D., Herath, S., and Musiake, K. (2000). Comparison of different distributed hydrological models for characterization of catchment spatial variability. *Hydrological Processes*, 14(3):403–416.

Yoshikawa, S., Cho, J., Yamada, H. G., Hanasaki, N., and Kanae, S. (2014). An assessment of global net irrigation water requirements from various water supply sources to sustain irrigation: rivers and reservoirs (1960–2050). *Hydrology and Earth System Sciences*, 18(10):4289–4310. doi:https://dx.doi.org/10.5194/hess-18-4289-2014.

You, J. and Cai, X. (2008). Hedging rule for reservoir operations: 1. A theoretical analysis. *Water Resources Research*, 44(1):n/a–n/a. doi:https://dx.doi.org/10.1029/2006WR005481.

Yue, S., Chen, M., Wen, Y., and Lu, G. (2015). Service-oriented model-encapsulation strategy for sharing and integrating heterogeneous geo-analysis models in an open web environment. *ISPRS Journal of Photogrammetry and Remote Sensing*. doi:https://dx.doi.org/10.1016/j.isprsjprs.2015.11.002.

Zhao, S., Peng, C., Jiang, H., Tian, D., Lei, X., and Zhou, X. (2006). Land use change in Asia and the ecological consequences. *Ecological Research*, 21(6):890–896. doi:https://dx.doi.org/10.1007/s11284-006-0048-2.

Zhu, Z., Bi, J., Pan, Y., Ganguly, S., Anav, A., Xu, L., Samanta, A., Piao, S., Nemani, R., and Myneni, R. (2013). Global data sets of vegetation Leaf Area Index (LAI)3g and Fraction of Photosynthetically Active Radiation (FPAR)3g derived from Global Inventory Modeling and Mapping Studies (GIMMS) Normalized Difference Vegetation Index (NDVI3g) for the period 1981 to 2011. *Remote Sensing*, 5(2):927–948. doi:https://dx.doi.org/10.3390/rs5020927.

Zou, G., Zhang, B., Zheng, J., Li, Y., and Ma, J. (2012). MaaS: model as a service in cloud computing and Cyber-I space. In *2012 IEEE 12th International Conference on*

Computer and Information Technology, pages 1125–1130. IEEE. doi:https://dx.doi.org/10.1109/CIT.2012.228.

Zulkafli, Z., Buytaert, W., Onof, C., Lavado, W., and Guyot, J. L. (2013). A critical assessment of the JULES land surface model hydrology for humid tropical environments. *Hydrology and Earth System Sciences*, 17(3):1113–1132. doi:https://dx.doi.org/10.5194/hess-17-1113-2013.

Zulkafli, Z. D. (2014). *The hydrology of the Peruvian Amazon river and its sensitivity to climate change*. PhD thesis, Imperial College London.

Appendices

A Manual for *lulcc*

Package 'lulcc'

July 28, 2016

Title Land Use Change Modelling in R

Version 1.0.0

Author Simon Moulds <simon.moulds10@imperial.ac.uk>

Maintainer Simon Moulds <simon.moulds10@imperial.ac.uk>

Description Classes and methods for spatially explicit land use change modelling in R.

Depends methods,
raster,
R (>= 3.1.0)

License GPL (>= 2)

LazyData true

Imports ROCR,
lattice,
rasterVis

Suggests caret,
rpart,
randomForest,
gsubfn,
Hmisc,
plyr,
RColorBrewer,

Collate 'class-PredictiveModelList.R'
'class-NeighbRasterStack.R'
'class-LulcRasterStack.R'
'class-ExpVarRasterStack.R'
'class-Model.R'
'class-ThreeMapComparison.R'
'AgreementBudget.R'
'ExpVarRasterStack.R'
'FigureOfMerit.R'
'LulcRasterStack.R'
'Model.R'
'NeighbRasterStack.R'
'class-PerformanceList.R'
'PerformanceList.R'
'class-PredictionList.R'
'PredictionList.R'

'as.data.frame.R'
'PredictiveModelList.R'
'ThreeMapComparison.R'
'allocate.R'
'allow.R'
'approxExtrapDemand.R'
'c.PredictiveModelList.R'
'compareAUC.R'
'crop.R'
'index.R'
'crossTabulate.R'
'data.R'
'getPredictiveModelInputData.R'
'length.R'
'lulcc-package.R'
'names.R'
'partition.R'
'plot.AgreementBudget.R'
'plot.FigureOfMerit.R'
'plot.Performance.R'
'plot.R'
'predict.R'
'resample.R'
'show.R'
'subset.R'
'summary.R'
'total.R'

NeedsCompilation yes

Repository CRAN

Date/Publication 2015-08-19 11:50:07

R topics documented:

lulcc-package . . . 3
AgreementBudget . . . 8
AgreementBudget-class . . . 9
allocate . . . 9
allow . . . 10
allowNeighb . . . 12
approxExtrapDemand . . . 13
as.data.frame.ExpVarRasterStack . . . 15
c.PredictiveModelList . . . 16
clue . . . 17
ClueModel . . . 18
ClueModel-class . . . 20
clues . . . 20
CluesModel . . . 21
CluesModel-class . . . 23
compareAUC . . . 23
ContinuousLulcRasterStack-class . . . 24
crossTabulate . . . 25

DiscreteLulcRasterStack-class . . . 26
ExpVarRasterStack . . . 27
ExpVarRasterStack-class . . . 28
Extract by index . . . 29
FigureOfMerit . . . 30
FigureOfMerit-class . . . 31
getPredictiveModelInputData . . . 31
LulcRasterStack . . . 32
LulcRasterStack-class . . . 33
Model-class . . . 34
NeighbRasterStack . . . 34
NeighbRasterStack-class . . . 36
ordered . . . 36
OrderedModel . . . 37
OrderedModel-class . . . 38
partition . . . 39
PerformanceList . . . 40
PerformanceList-class . . . 41
pie . . . 41
plot . . . 42
plot.AgreementBudget . . . 43
plot.FigureOfMerit . . . 44
plot.PerformanceList . . . 45
predict.PredictiveModelList . . . 46
PredictionList . . . 47
PredictionList-class . . . 48
PredictiveModelList . . . 48
PredictiveModelList-class . . . 49
resample,ExpVarRasterStack,Raster-method . . . 49
roundSum . . . 50
show,PredictiveModelList-method . . . 51
sibuyan . . . 51
subset,PredictiveModelList-method . . . 52
summary . . . 54
ThreeMapComparison . . . 54
ThreeMapComparison-class . . . 55
total . . . 56
updateDataFrame . . . 57

Index **58**

`lulcc-package` *lulcc: land use change modelling in R*

Description

The lulcc package is an open and extensible framework for land use change modelling in R.

Details

The aims of the package are as follows:

1. to improve the reproducibility of scientific results and encourage reuse of code within the land use change modelling community
2. to make it easy to directly compare and combine different model structures
3. to allow users to perform several aspects of the modelling process within the same environment

To achieve these aims the package utilises an object-oriented approach based on the S4 system, which provides a formal structure for the modelling framework. Generic methods implemented for the `lulcc` classes include `summary`, `show`, and `plot`.

Land use change models are represented by objects inheriting from the superclass `Model`. This class is designed to represent general information required by all models while specific models are represented by its subclasses. Currently the package includes two discrete land use change models: an implementation of the Change in Land Use and its Effects at Small Regional extent (CLUE-S) model (Verburg et al., 2002) (class `CluesModel`) and an ordered procedure based on the algorithm described by Fuchs et al. (2013) but modified to allow stochastic transitions (class `OrderedModel`). An implementation of the continuous land use change model CLUE (Veldkamp and Fresco, 1996; Verburg and Bouma, 1999) is also included.

The main input to inductive land use change models is a set of predictive models relating observed land use or land use change to spatially explicit explanatory variables. A predictive model is usually obtained for each category or transition. In lulcc these models are represented by the class `PredictiveModelList`. Currently lulcc supports binary logistic regression, provided by base R (`glm`), recursive partitioning and regression trees, provided by package `rpart` and random forest, provided by package `randomForest`. To a large extent the success of the allocation routine depends on the strength of the predictive models.

To validate model output lulcc includes a method developed by Pontius et al. (2011) that simultaneously compares a reference map for time 1, a reference map for time 2 and a simulated map for time 2 at multiple resolutions. In lulcc the results of the comparison are represented by the class `ThreeMapComparison`. From objects of this class it is straightforward to extract information about different sources of agreement and disagreement, represented by the class `AgreementBudget`, which can then be plotted. The results of the comparison are conveniently summarised by the figure of merit, represented by the class `FigureOfMerit`.

In addition to the core functionality described above, lulcc inludes several utility functions to assist with the model building process. Two example datasets are also included.

Author(s)

Simon Moulds

References

Fuchs, R., Herold, M., Verburg, P.H., and Clevers, J.G.P.W. (2013). A high-resolution and harmonized model approach for reconstructing and analysing historic land changes in Europe, Biogeosciences, 10:1543-1559.

Pontius Jr, R.G., Peethambaram, S., Castella, J.C. (2011). Comparison of three maps at multiple resol utions: a case study of land change simulation in Cho Don District, Vietnam. Annals of the Association of American Geographers 101(1): 45-62.

Veldkamp, A., & Fresco, L. O. (1996). CLUE-CR: an integrated multi-scale model to simulate land use change scenarios in Costa Rica. Ecological modelling, 91(1), 231-248.

Verburg, P.H., & Bouma, J. (1999). Land use change under conditions of high population pressure: the case of Java. Global environmental change, 9(4), 303-312.

Verburg, P.H., Soepboer, W., Veldkamp, A., Limpiada, R., Espaldon, V., Mastura, S.S. (2002). Modeling the spatial dynamics of regional land use: the CLUE-S model. Environmental management, 30(3):391-405.

Examples

```
## Not run:

## Plum Island Ecosystems

data(pie)

## Observed maps
lu <- DiscreteLulcRasterStack(x=stack(pie[1:3]),
                              categories=c(1,2,3),
                              labels=c("Forest","Built","Other"),
                              t=c(0,6,14))
plot(lu)

crossTabulate(x=lu, times=c(0,14))

## Explanatory variables
idx <- data.frame(var=c("ef_001","ef_002","ef_003"),
                  yr=c(0,0,0),
                  dynamic=c(FALSE,FALSE,FALSE))
idx

ef <- ExpVarRasterStack(x=stack(pie[4:6]), index=idx)

part <- partition(x=lu, size=0.1, spatial=TRUE, t=0)
train.data <- getPredictiveModelInputData(lu=lu,
                                          ef=ef,
                                          cells=part[["train"]],
                                          t=0)

## predictive modelling
forest.form <- as.formula("Forest ~ ef_001 + ef_002")
built.form <- as.formula("Built ~ ef_001 + ef_002 + ef_003")
other.form <- as.formula("Other ~ ef_001 + ef_002")

library(randomForest)
library(rpart)

forest.glm <- glm(forest.form, family=binomial, data=train.data)
forest.rprt <- rpart(forest.form, data=train.data)
forest.rf <- randomForest(forest.form, method="class", data=train.data)

built.glm <- glm(built.form, family=binomial, data=train.data)
built.rprt <- rpart(built.form, data=train.data)
built.rf <- randomForest(built.form, method="class", data=train.data)

other.glm <- glm(other.form, family=binomial, data=train.data)
other.rprt <- rpart(other.form, data=train.data)
other.rf <- randomForest(other.form, method="class", data=train.data)
```

```
## Binomial logistic regression
glm.mods <- PredictiveModelList(list(forest.glm, built.glm, other.glm),
                                categories=lu@categories,
                                labels=lu@labels)

## Recursive partitioning and regression trees
rprt.mods <- PredictiveModelList(list(forest.rprt, built.rprt, other.rprt),
                                 categories=lu@categories,
                                 labels=lu@labels)

## Random forests
rf.mods <- PredictiveModelList(list(forest.rf, built.rf, other.rf),
                               categories=lu@categories,
                               labels=lu@labels)

test.data <- getPredictiveModelInputData(lu=lu,
                                         ef=ef,
                                         cells=part[["test"]],
                                         t=0)

glm.pred <- PredictionList(models=glm.mods, newdata=test.data)
glm.perf <- PerformanceList(pred=glm.pred, measure="rch")

rprt.pred <- PredictionList(models=rprt.mods, newdata=test.data)
rprt.perf <- PerformanceList(pred=rprt.pred, measure="rch")

rf.pred <- PredictionList(models=rf.mods, newdata=test.data)
rf.perf <- PerformanceList(pred=rf.pred, measure="rch")

p <- plot(list(glm=glm.perf, rpart=rprt.perf, rf=rf.perf))

## Probability maps
all.data <- as.data.frame(x=ef, cells=part[["all"]])
probmaps <- predict(object=glm.mods,
                    newdata=all.data,
                    data.frame=TRUE)

points <- rasterToPoints(lu[[1]], spatial=TRUE)
probmaps <- SpatialPointsDataFrame(points, probmaps)
probmaps <- rasterize(x=probmaps, y=lu[[1]],
                      field=names(probmaps))

p <- levelplot(probmaps, layout=c(2,2), margin=FALSE)

## Demand scenario
dmd <- approxExtrapDemand(lu=lu, tout=0:14)

## CLUE-S modelling
clues.model <- CluesModel(observed.lulc=lu,
                          explanatory.variables=ef,
                          predictive.models=glm.mods,
                          time=0:14,
                          demand=dmd,
                          history=NULL,
                          mask=NULL,
                          neighbourhood=NULL,
```

```
                            transition.rules=matrix(data=1, nrow=3, ncol=3),
                            neighbourhood.rules=NULL,
                            elasticity=c(0.2,0.2,0.2),
                            iteration.factor=0.00001,
                            max.iteration=1000,
                            max.difference=5,
                            ave.difference=5)

clues.result <- allocate(clues.model)

## Ordered modelling
ordered.model <- OrderedModel(observed.lulc=lu,
                              explanatory.variables=ef,
                              predictive.models=glm.mods,
                              time=0:14,
                              demand=dmd,
                              transition.rules=matrix(data=1, 3, 3),
                              order=c(2,1,3))

ordered.result <- allocate(ordered.model, stochastic=FALSE)

## Validation

clues.tabs <- ThreeMapComparison(x=lu[[1]],
                                 x1=lu[[3]],
                                 y1=clues.result[[15]],
                                 factors=2^(1:8),
                                 categories=lu@categories,
                                 labels=lu@labels)

clues.agr <- AgreementBudget(x=clues.tabs)
clues.fom <- FigureOfMerit(x=clues.tabs)
ordered.tabs <- ThreeMapComparison(x=lu[[1]],
                                 x1=lu[[3]],
                                 y1=ordered.result[[15]],
                                 factors=2^(1:8),
                                 categories=lu@categories,
                                 labels=lu@labels)

ordered.agr <- AgreementBudget(x=ordered.tabs)
ordered.fom <- FigureOfMerit(x=ordered.tabs)

p1 <- plot(clues.agr, from=1, to=2)
p2 <- plot(ordered.agr, from=1, to=2)

agr.p <- c("CLUE-S"=p1, Ordered=p2, layout=c(1,2))
agr.p

p1 <- plot(clues.fom, from=1, to=2)
p2 <- plot(ordered.fom, from=1, to=2)

fom.p <- c("CLUE-S"=p1, Ordered=p2, layout=c(1,2))
fom.p


## End(Not run)
```

AgreementBudget *Create an AgreementBudget object*

Description

This function quantifies sources of agreement and disagreement between a reference map for time 1, a reference map for time 2 and a simulated map for time 2 to provide meaningful information about the performance of land use change simulations.

Usage

```
AgreementBudget(x, ...)

## S4 method for signature ThreeMapComparison
AgreementBudget(x, ...)

## S4 method for signature RasterLayer
AgreementBudget(x, ...)
```

Arguments

x	a ThreeMapComparison object or RasterLayer
...	additional arguments passed to ThreeMapComparison

Details

The types of agreement and disagreement considered are those descibed in Pontius et al. (2011):

1. Persistence simulated correctly (agreement)
2. Persistence simulated as change (disagreement)
3. Change simulated incorrectly (disagreement)
4. Change simulated correctly (agreement)
5. Change simulated as persistence (disagreement)

Value

An AgreementBudget object.

References

Pontius Jr, R.G., Peethambaram, S., Castella, J.C. (2011). Comparison of three maps at multiple resolutions: a case study of land change simulation in Cho Don District, Vietnam. Annals of the Association of American Geographers 101(1): 45-62.

See Also

AgreementBudget-class, plot.AgreementBudget, ThreeMapComparison, FigureOfMerit

Examples

```
## see lulcc-package examples
```

AgreementBudget-class *Class AgreementBudget*

Description

An S4 class for information about sources of agreement and disagreement between three categorical raster maps.

Slots

`tables` list of data.frames that depict the three dimensional table described by Pontius et al. (2011) at different resolutions

`factors` numeric vector of aggregation factors

`maps` list of RasterStack objects containing land use maps at different resolutions

`categories` numeric vector of land use categories

`labels` character vector corresponding to `categories`

`overall` data.frame containing the overall agreement budget

`category` list of data.frames showing the agreement budget for each category

`transition` list of data.frames showing the agreement budget for all possible transitions

allocate *Allocate land use change spatially*

Description

Perform spatial allocation of land use change using different models. Currently the function provides an implementation of the Change in Land Use and its Effects (CLUE; Veldkamp and Fresco, 1996, Verburg et al., 1996), CLUE at Small regional extent (CLUE-S; Verburg et al., 2002) and an ordered procedure based on the algorithm described by Fuchs et al., (2013), modified to allow stochastic transitions.

Usage

```
allocate(model, ...)

## S4 method for signature CluesModel
allocate(model, ...)

## S4 method for signature ClueModel
allocate(model, ...)

## S4 method for signature OrderedModel
allocate(model, stochastic = TRUE, ...)
```

Arguments

model	an object inheriting from class Model
stochastic	logical
...	additional arguments for specific methods

Value

LulcRasterStack.

References

Fuchs, R., Herold, M., Verburg, P.H., and Clevers, J.G.P.W. (2013). A high-resolution and harmonized model approach for reconstructing and analysing historic land changes in Europe, Biogeosciences, 10:1543-1559.

Veldkamp, A., & Fresco, L. O. (1996). CLUE-CR: an integrated multi-scale model to simulate land use change scenarios in Costa Rica. Ecological modelling, 91(1), 231-248.

Verburg, P.H., & Bouma, J. (1999). Land use change under conditions of high population pressure: the case of Java. Global environmental change, 9(4), 303-312.

Verburg, P.H., Soepboer, W., Veldkamp, A., Limpiada, R., Espaldon, V., Mastura, S.S. (2002). Modeling the spatial dynamics of regional land use: the CLUE-S model. Environmental management, 30(3):391-405.

See Also

CluesModel

Examples

```
## see lulcc-package examples
```

---

allow *Implement decision rules for land use change*

Description

Identify legitimate transitions based on land use history and specific transition rules.

Usage

```
allow(x, categories, cd, rules, hist = NULL, ...)
```

Arguments

x	numeric vector containing the land use pattern for the current timestep
categories	numeric vector containing land use categories in the study region
cd	numeric vector indicating the direction of change for each land use category. A value of 1 means demand is increasing (i.e. the number of cells belonging to the category must increase), -1 means decreasing demand and 0 means demand is static

rules	matrix. See details
hist	numeric vector containing land use history (values represent the number of timesteps the cell has contained the current land use category). Only required for rules 2 and 3
...	additional arguments (none)

Details

Decision rules are based on those described by Verburg et al. (2002). The rules input argument is a square matrix with dimensions equal to the number of land use categories in the study region where rows represent the current land use and columns represent future transitions. The value of each element should represent a rule from the following list:

1. rule == 0 | rule == 1: this rule concerns specific land use transitions that are allowed (1) or not (0)
2. rule > 100 & rule < 1000: this rule imposes a time limit (rule - 100) on land use transitions, after which land use change is not allowed. Time is taken from hist
3. rule > 1000: this rule imposes a minimum period of time (rule-1000) before land use is allowed to change

allow should be called from allocate methods. The output is a matrix with the same dimensions as the matrix used internally by allocation functions to store land use suitability. Thus, by multiplying the two matrices together, disallowed transitions are removed from the allocation procedure.

Value

A matrix.

References

Verburg, P.H., Soepboer, W., Veldkamp, A., Limpiada, R., Espaldon, V., Mastura, S.S. (2002). Modeling the spatial dynamics of regional land use: the CLUE-S model. Environmental management, 30(3):391-405.

See Also

allowNeighb

Examples

```
## Plum Island Ecosystems

## load observed land use maps
lu <- DiscreteLulcRasterStack(x=stack(pie[1:3]),
                              categories=c(1,2,3),
                              labels=c("Forest","Built","Other"),
                              t=c(0,6,14))

## get land use values
x <- getValues(lu[[1]])
x <- x[!is.na(x)]

## create vector of arbitrary land use history values
hist <- sample(1:10, length(x), replace=TRUE)
```

```
## calculate demand and get change direction for first timestep
dmd <- approxExtrapDemand(lu=lu, tout=0:14)
cd <- dmd[2,] - dmd[1,]

## create rules matrix, only allowing forest to change if the cell has
## belonged to forest for more than 8 years
rules <- matrix(data=c(1,1008,1008,
                       1,1,1,
                       1,1,1), nrow=3, ncol=3, byrow=TRUE)

allow <- allow(x=x,
               hist=hist,
               categories=lu@categories,
               cd=cd,
               rules=rules)

## create raster showing cells that are allowed to change from forest to built
r <- lu[[1]]
r[!is.na(r)] <- allow[,2]
r[lu[[1]] != 1] <- NA
plot(r)

## NB output is only useful when used within allocation routine
```

`allowNeighb` *Implement neighbourhood decision rules*

Description

Identify legitimate transitions for each cell according to neighbourhood decision rules.

Usage

```
allowNeighb(neighb, x, categories, rules, ...)
```

Arguments

`neighb`	a NeighbRasterStack object
`x`	a categorical RasterLayer to which neighbourhood rules should be applied. If `neighb` is supplied it is updated with this map
`categories`	numeric vector containing land use categories. If `allowNeighb` is called from an allocation model this argument should contain all categories in the simulation, regardless of whether they're associated with a neighbourhood decision rule
`rules`	a numeric vector with neighbourhood decision rules. Each rule is a value between 0 and 1 representing the threshold neighbourhood value above which change is allowed. Rules should correspond with `x@categories`
`...`	additional arguments (none)

Value

A matrix.

See Also

allow, NeighbRasterStack

Examples

```
## Plum Island Ecosystems

## load observed land use maps
lu <- DiscreteLulcRasterStack(x=stack(pie[1:3]),
                              categories=c(1,2,3),
                              labels=c("Forest","Built","Other"),
                              t=c(0,6,14))

## create a NeighbRasterStack object for forest only
w <- matrix(data=1, nrow=3, ncol=3)
nb <- NeighbRasterStack(x=lu[[1]], weights=w, categories=1)

## only allow change to forest within neighbourhood of current forest cells
## note that rules can be any value between zero (less restrictive) and one
## (more restrictive)
nb.allow <- allowNeighb(neighb=nb,
                        x=lu[[1]],
                        categories=lu@categories,
                        rules=0.5)

## create raster showing cells allowed to change to forest
r <- lu[[1]]
r[!is.na(r)] <- nb.allow[,1]
plot(r)

## NB output is only useful when used within an allocation routine
```

---

approxExtrapDemand *Extrapolate land use area in time*

Description

Extrapolate land use area from two or more observed land use maps to provide a valid (although not necessarily realistic) demand scenario.

Usage

```
approxExtrapDemand(lu, ...)

## S4 method for signature LulcRasterStack
approxExtrapDemand(lu, tout, ...)

## S4 method for signature DiscreteLulcRasterStack
approxExtrapDemand(lu, tout, ...)
```

Arguments

lu	an LulcRasterStack object containing at least two maps
tout	numeric vector specifying the timesteps where interpolation is to take place. Comparable to the xout argument of Hmisc::approxExtrap
...	additional arguments to Hmisc::approxExtrap

Details

Many allocation routines, including the two included with lulcc, require non-spatial estimates of land use demand for every timestep in the study period. Some routines are coupled to complex economic models that predict future or past land use demand based on economic considerations; however, linear extrapolation of trends remains a useful technique.

Value

A matrix.

See Also

Hmisc::approxExtrap

Examples

```
## Plum Island Ecosystems

## load observed land use maps
lu <- DiscreteLulcRasterStack(x=stack(pie[1:3]),
                              categories=c(1,2,3),
                              labels=c("Forest","Built","Other"),
                              t=c(0,6,14))

## obtain demand scenario by interpolating between observed maps
dmd <- approxExtrapDemand(lu=lu, tout=0:14)

## plot
matplot(dmd, type="l", ylab="Demand (no. of cells)", xlab="Time point",
        lty=1, col=c("Green","Red","Blue"))
legend("topleft", legend=lu@labels, col=c("Green","Red","Blue"), lty=1)

## linear extrapolation is also possible
dmd <- approxExtrapDemand(lu=lu, tout=c(0:50))

## plot
matplot(dmd, type="l", ylab="Demand (no. of cells)", xlab="Time point",
        lty=1, col=c("Green","Red","Blue"))
legend("topleft", legend=lu@labels, col=c("Green","Red","Blue"), lty=1)
```

`as.data.frame.ExpVarRasterStack` *Coerce objects to data.frame*

Description

This function extracts data from all raster objects in `LulcRasterStack` or `ExpVarRasterStack` objects for a specified timestep.

Usage

```
## S3 method for class ExpVarRasterStack
as.data.frame(x, row.names = NULL,
  optional = FALSE, cells, t, ...)

## S3 method for class DiscreteLulcRasterStack
as.data.frame(x, row.names = NULL,
  optional = FALSE, cells, t, ...)

## S3 method for class ContinuousLulcRasterStack
as.data.frame(x, row.names = NULL,
  optional = FALSE, cells, t, ...)

## S4 method for signature ExpVarRasterStack
as.data.frame(x, row.names = NULL,
  optional = FALSE, cells, t, ...)

## S4 method for signature DiscreteLulcRasterStack
as.data.frame(x, row.names = NULL,
  optional = FALSE, cells, t, ...)

## S4 method for signature ContinuousLulcRasterStack
as.data.frame(x, row.names = NULL,
  optional = FALSE, cells, t, ...)
```

Arguments

`x`	an ExpVarRasterStack or LulcRasterStack object
`row.names`	NULL or a character vector giving the row.names for the data.frame. Missing values are not allowed
`optional`	logical. If TRUE, setting row names and converting column names (to syntactic names: see make.names) is optional
`cells`	index of cells to be extracted, which may be a `SpatialPoints*` object or a numeric vector representing cell numbers (see `raster::extract`)
`t`	numeric indicating the time under consideration
`...`	additional arguments (none)

Details

If x is a DiscreteLulcRasterStack object the raster corresponding to t is first transformed to a RasterBrick with a boolean layer for each class with `raster::layerize`.

Value

A data.frame.

See Also

as.data.frame, LulcRasterStack, ExpVarRasterStack, partition

Examples

```
## Not run:

## Plum Island Ecosystems

## load observed land use maps
lu <- DiscreteLulcRasterStack(x=stack(pie[1:3]),
                              categories=c(1,2,3),
                              labels=c("Forest","Built","Other"),
                              t=c(0,6,14))

## explanatory variables
idx <- data.frame(var=c("ef_001","ef_002","ef_003"),
                  yr=c(0,0,0),
                  dynamic=c(FALSE,FALSE,FALSE))

ef <- ExpVarRasterStack(x=stack(pie[4:6]), index=idx)

## separate data into training and testing partitions
part <- partition(x=lu[[1]], size=0.1, spatial=TRUE)
df1 <- as.data.frame(x=lu, cells=part[["all"]], t=0)
df2 <- as.data.frame(x=ef, cells=part[["all"]], t=0)
df <- cbind(df1,df2)


## End(Not run)
```

---

c.PredictiveModelList *Merge PredictiveModelList objects*

Description

Combine different PredictiveModelList objects into one

Usage

```
## S3 method for class PredictiveModelList
c(..., recursive = FALSE)
```

Arguments

...	two or more PredictiveModelList objects
recursive	for consistency with generic method (ignored)

Value

a PredictiveModelList object

Examples

```
## Not run:

## Plum Island Ecosystems

## load data
data(pie)

## observed maps
obs <- LulcRasterStack(x=pie,
                   pattern="lu",
                   categories=c(1,2,3),
                   labels=c("Forest","Built","Other"),
                   t=c(0,6,14))

## explanatory variables
ef <- ExpVarRasterStack(x=pie, pattern="ef")

part <- partition(x=obs[[1]], size=0.1, spatial=TRUE)
train.data <- getPredictiveModelInputData(obs=obs, ef=ef, cells=part[["train"]], t=0)

forms <- list(Built ~ ef_001+ef_002+ef_003,
              Forest ~ 1,
              Other ~ ef_001+ef_002)

glm.models <- glmModels(formula=forms, family=binomial, data=train.data, obs=obs)
glm.models

## separate glm.models into two PredictiveModelList objects
mod1 <- glm.models[[1]]
mod2 <- glm.models[[2:3]]

## put them back together again
glm.models <- c(mod1, mod2)
glm.models


## End(Not run)
```

---

clue *CLUE*

Description

Allocate land use change using the CLUE algorithm.

Usage

```
clue(lu0.vals, regr, demand, elasticity, change.rule, min.elasticity,
  max.elasticity, min.change, max.change, min.value, max.value, max.iteration,
  max.difference, cell.area, ncell, ncode)
```

Arguments

lu0.vals	matrix containing non-NA values from lu0
regr	matrix containing...
demand	matrix with demand for each land use category in terms of number of cells to be allocated. The first row should be the number of cells allocated to the initial land use map, the second row should be the number of cells to allocate in the subsequent time point
elasticity	Initial elasticity value. Default is 0.1
change.rule	numeric vector specifying for each land use whether change is allowed in either direction (0), allowed in the direction of demand only (-1) or not allowed (1)
min.elasticity	Minimum elasticity value. Default is 0.001
max.elasticity	Maximum elasticity value. Default is 1.5
min.change	numeric vector indicating for each land use the minimum amount of change that is allowed to occur in one time step
max.change	numeric vector indicating for each land use the maximum amount of change that is allowed to occur in one time step
min.value	numeric vector indicating the minimum fraction of each land use in a given cell
max.value	numeric vector indicating the maximum fraction of each land use in a given cell
max.iteration	The maximum number of iterations allowed at each time step
max.difference	The maximum allowable difference between demand and allocated area
cell.area	The area of each grid cell in the study region, which should have the same units as the demand
ncell	number of cells considered for change (equal to the length of lu0.vals
ncode	number of land use categories under consideration
...	additional arguments (none)

Value

numeric vector with updated land use values.

Examples

```
## See lulcc-package examples
```

ClueModel *Create a ClueModel object*

Description

Methods to create a ClueModel object to supply to allocate.

Usage

```
ClueModel(observed.lulc, explanatory.variables, predictive.models, time, demand,
  elasticity = 0.1, change.rule, min.elasticity = 0.001,
  max.elasticity = 1.5, min.value, max.value, min.change, max.change,
  max.iteration = 1000, max.difference, cell.area)
```

Arguments

observed.lulc	an LulcRasterStack
explanatory.variables	an ExpVarRasterStack object
predictive.models	a PredictiveModelList object
time	numeric vector containing timesteps over which simulation will occur
demand	matrix with demand for each land use category in terms of number of cells to be allocated. The first row should be the number of cells allocated to the initial observed land use map (i.e. the land use map for time 0)
elasticity	Initial elasticity value. Default is 0.1
change.rule	numeric vector specifying for each land use whether change is allowed in either direction (0), allowed in the direction of demand only (-1) or not allowed (1)
min.elasticity	Minimum elasticity value. Default is 0.001
max.elasticity	Maximum elasticity value. Default is 1.5
min.value	numeric vector indicating the minimum fraction of each land use in a given cell
max.value	numeric vector indicating the maximum fraction of each land use in a given cell
min.change	numeric vector indicating for each land use the minimum amount of change that is allowed to occur in one time step
max.change	numeric vector indicating for each land use the maximum amount of change that is allowed to occur in one time step
max.iteration	The maximum number of iterations allowed at each time step
max.difference	The maximum allowable difference between demand and allocated area
cell.area	The area of each grid cell in the study region, which should have the same units as the demand
...	additional arguments (none)

Value

A ClueModel object.

References

Veldkamp, A., & Fresco, L. O. (1996). CLUE-CR: an integrated multi-scale model to simulate land use change scenarios in Costa Rica. Ecological modelling, 91(1), 231-248.

Verburg, P.H., & Bouma, J. (1999). Land use change under conditions of high population pressure: the case of Java. Global environmental change, 9(4), 303-312.

See Also

ClueModel-class, allocate

Examples

```
## see lulcc-package examples
```

ClueModel-class *Class ClueModel*

Description

An S4 class to represent inputs to the CLUE land use change model.

Slots

`observed.lulc` a ContinuousLulcRasterStack object

`explanatory.variables` an ExpVarRasterStack object

`predictive.models` a PredictiveModelList object

`time` numeric vector of timesteps over which simulation will occur

`demand` matrix containing demand scenario

`elasticity` numeric

`change.rule` numeric

`min.elasticity` numeric

`max.elasticity` numeric

`min.value` numeric

`max.value` numeric

`min.change` numeric

`max.change` numeric

`max.iteration` numeric

`max.difference` numeric

`cell.area` numeric

`categories` numeric vector of land use categories

`labels` character vector corresponding to `categories`

clues *CLUE-S*

Description

Allocate land use change using the CLUE-S algorithm.

Usage

```
clues(lu0, lu0.vals, tprob, nb = NULL, nb.rules = NULL,
  transition.rules = NULL, hist.vals = NULL, mask.vals = NULL, demand,
  categories, elasticity, iteration.factor, max.iteration, max.difference,
  ave.difference, ...)
```

Arguments

lu0	RasterLayer showing initial land use
lu0.vals	numeric containing non-NA values from lu0
tprob	matrix with land use suitability values. Columns should correspond to categories, rows should correspond with cells
nb	neighbourhood map. See CluesModel
nb.rules	neighbourhood rules. See CluesModel documentation
transition.rules	transition rules. See CluesModel documentation
hist.vals	numeric vector detailing the number of consecutive time steps each cell has been allocated to its current land use
mask.vals	numeric vector containing binary values where 0 indicates cells that are not allowed to change
demand	matrix with demand for each land use category in terms of number of cells to be allocated. The first row should be the number of cells allocated to the initial land use map, the second row should be the number of cells to allocate in the subsequent time point
categories	numeric vector containing land use categories
elasticity	elasticity values. See CluesModel documentation
iteration.factor	iteration factor. See CluesModel documentation
max.iteration	The maximum number of iterations allowed at each time step
max.difference	The maximum allowable difference between demand and allocated area
ave.difference	The maximum allowable average difference across all land uses
...	additional arguments (none)

Value

numeric vector with updated land use values.

Examples

```
## See lulcc-package examples
```

CluesModel *Create a CluesModel object*

Description

Methods to create a CluesModel object to supply to allocate.

Usage

```
CluesModel(observed.lulc, explanatory.variables, predictive.models, time,
  demand, history = NULL, mask = NULL, neighbourhood = NULL,
  transition.rules, neighbourhood.rules = NULL, elasticity,
  iteration.factor = 1e-05, max.iteration = 1000, max.difference = 5,
  ave.difference = 5, ...)
```

Arguments

observed.lulc	an LulcRasterStack
explanatory.variables	an ExpVarRasterStack object
predictive.models	a PredictiveModelList object
time	numeric vector containing timesteps over which simulation will occur
demand	matrix with demand for each land use category in terms of number of cells to be allocated. The first row should be the number of cells allocated to the initial observed land use map (i.e. the land use map for time 0)
history	RasterLayer containing land use history (values represent the number of years the cell has contained the current land use category)
mask	RasterLayer containing binary values where 0 indicates cells that are not allowed to change
neighbourhood	an object of class NeighbRasterStack
transition.rules	matrix with land use change decision rules
neighbourhood.rules	numeric with neighbourhood decision rules
elasticity	numeric indicating the elasticity of each land use category to change. Elasticity varies between 0 and 1, with 0 indicating a low resistance to change and 1 indicating a high resistance to change
iteration.factor	TODO,
max.iteration	The maximum number of iterations allowed at each time step
max.difference	The maximum allowable difference between demand and allocated area
ave.difference	The maximum allowable average difference across all land uses
...	additional arguments (none)

Value

A CluesModel object.

References

Verburg, P.H., Soepboer, W., Veldkamp, A., Limpiada, R., Espaldon, V., Mastura, S.S. (2002). Modeling the spatial dynamics of regional land use: the CLUE-S model. Environmental management, 30(3):391-405.

See Also

CluesModel-class, allocate

Examples

```
## see lulcc-package examples
```

CluesModel-class *Class CluesModel*

Description

An S4 class to represent inputs to the CLUE-S land use change model.

Slots

`observed.lulc` an LulcRasterStack object

`explanatory.variables` an ExpVarRasterStack object

`predictive.models` a PredictiveModelList object

`time` numeric vector of timesteps over which simulation will occur

`demand` matrix containing demand scenario

`history` RasterLayer showing land use history or NULL

`mask` RasterLayer showing masked areas or NULL

`neighbourhood` NeighbRasterStack object or NULL

`transition.rules` matrix with land use change decision rules

`neighbourhood.rules` numeric with neighbourhood decision rules

`elasticity` numeric indicating elasticity to change (only required for

`iteration.factor` TODO

`max.iteration` TODO

`max.difference` TODO

`ave.difference` TODO

`categories` numeric vector of land use categories

`labels` character vector corresponding to `categories`

compareAUC *Calculate the area under the ROC curve (AUC)*

Description

Estimate the AUC for each `ROCR::prediction` object in a `PredictionList` object.

Usage

```
compareAUC(pred, ...)

## S4 method for signature PredictionList
compareAUC(pred, digits = 4, ...)

## S4 method for signature list
compareAUC(pred, digits = 4, ...)
```

Arguments

pred	a PredictionList object or a list of these
digits	numeric indicating the number of digits to be displayed after the decimal point for AUC values
...	additional arguments (none)

Details

The user can compare the performance of different statistical models by providing a list of PredictionList objects. Note that compareAUC should be used in conjunction with other comparison methods because the AUC does not contain as much information as, for instance, the ROC curve itself (Pontius and Parmentier, 2014).

Value

A data.frame.

References

Sing, T., Sander, O., Beerenwinkel, N., Lengauer, T. (2005). ROCR: visualizing classifier performance in R. Bioinformatics 21(20):3940-3941.

Pontius Jr, R. G., & Parmentier, B. (2014). Recommendations for using the relative operating characteristic (ROC). Landscape ecology, 29(3), 367-382.

See Also

PredictionList, ROCR::performance

Examples

```
## see PredictiveModelList examples
```

ContinuousLulcRasterStack-class

Class ContinuousLulcRasterStack

Description

A virtual S4 class for observed land use maps.

Slots

filename see raster::Raster-class

layers see raster::Raster-class

title see raster::Raster-class

extent see raster::Raster-class

rotated see raster::Raster-class

rotation see raster::Raster-class

ncols see raster::Raster-class

nrows see raster::Raster-class

crs see raster::Raster-class

history see raster::Raster-class

z see raster::Raster-class

t numeric vector with timesteps corresponding to each observed map

categories numeric vector of land use categories

labels character vector corresponding to categories

---

crossTabulate *Cross tabulate land use transitions*

Description

Cross tabulate land use transitions using raster::crosstab. This step should form the basis of further research into the processes driving the most important transitions in the study region (Pontius et al., 2004).

Usage

```
crossTabulate(x, y, ...)

## S4 method for signature RasterLayer,RasterLayer
crossTabulate(x, y, categories,
  labels = as.character(categories), ...)

## S4 method for signature DiscreteLulcRasterStack,ANY
crossTabulate(x, y, times, ...)
```

Arguments

x	RasterLayer representing land use map from an earlier timestep or an LulcRasterStack object containing at least two land use maps for different points in time
y	RasterLayer representing land use map from a later timestep. Not used if x is an LulcRasterStack object
categories	numeric vector containing land use categories to consider. Not used if x is an LulcRasterStack object
labels	character vector (optional) with labels corresponding to categories. Not used if x is an LulcRasterStack object
times	numeric vector representing the time points of two land use maps from LulcRasterStack
...	additional arguments to raster::crosstab

Value

A data.frame.

References

Pontius Jr, R.G., Shusas, E., McEachern, M. (2004). Detecting important categorical land changes while accounting for persistence. Agriculture, Ecosystems & Environment 101(2):251-268.

See Also

LulcRasterStack, raster::crosstab

Examples

```
## Not run:

## Plum Island Ecosystems

## load observed land use maps
lu <- DiscreteLulcRasterStack(x=stack(pie[1:3]),
                              categories=c(1,2,3),
                              labels=c("Forest","Built","Other"),
                              t=c(0,6,14))

crossTabulate(x=lu, times=c(0,6))
crossTabulate(x=lu, times=c(0,14))

## RasterLayer input
crossTabulate(x=lu[[1]],
              y=lu[[3]],
              categories=c(1,2,3),
              labels=c("forest","built","other"))


## End(Not run)
```

DiscreteLulcRasterStack-class

Class DiscreteLulcRasterStack

Description

A virtual S4 class for observed land use maps.

Slots

filename see raster::Raster-class

layers see raster::Raster-class

title see raster::Raster-class

extent see raster::Raster-class

rotated see raster::Raster-class

rotation see raster::Raster-class

ncols see raster::Raster-class

nrows see raster::Raster-class

`crs` see `raster::Raster-class`

`history` see `raster::Raster-class`

`z` see `raster::Raster-class`

`t` numeric vector with timesteps corresponding to each observed map

`categories` numeric vector of land use categories

`labels` character vector corresponding to `categories`

`ExpVarRasterStack` *Create an ExpVarRasterStack object*

Description

Methods to create an ExpVarRasterStack object, which may be created from file or an existing Raster* object.

Usage

```
ExpVarRasterStack(x, ...)

## S4 method for signature character
ExpVarRasterStack(x, ...)

## S4 method for signature RasterStack
ExpVarRasterStack(x, index, ...)
```

Arguments

`x`	Raster* object
`index`	data.frame
`...`	additional arguments to `raster::stack`

Details

Inductive and deductive land use change models predict the allocation of change based on spatially explicit biophysical and socioeconomic covariates. These may be static, such as elevation or geology, or dynamic, such as maps of population density or road networks. To identify whether a covariable is static or dynamic a data frame is supplied to the ExpVarRasterStack constructor function with three columns: the first column specifies the name of the variable, the second column specifies the time point for which it is relevant and the third column specifies whether it is dynamic or not. Data frame rows should correspond to the individual layers of the RasterStack object containing the explanatory variables. If dynamic variables are used it is not necessary to supply a map for each time point in the simulation: during allocation the most recent map will automatically be selected.

Value

An ExpVarRasterStack object.

See Also

raster::stack

Examples

```
## Plum Island Ecosystems

idx <- data.frame(var=paste("ef_", formatC(1:3, width=3, flag=0)),
                  yr=rep(0,3),
                  dynamic=rep(FALSE,3))

ef <- ExpVarRasterStack(x=stack(pie[4:6]), index=idx)

## Sibuyan

idx <- data.frame(var=paste("ef_", formatC(1:13, width=3, flag=0)),
                  yr=rep(0,13),
                  dynamic=rep(FALSE,13))

ef <- ExpVarRasterStack(x=stack(sibuyan$maps[3:15]), index=idx)
```

ExpVarRasterStack-class

Class ExpVarRasterStack

Description

An S4 class for explanatory variables.

Slots

filename see raster::Raster-class

layers see raster::Raster-class

title see raster::Raster-class

extent see raster::Raster-class

rotated see raster::Raster-class

rotation see raster::Raster-class

ncols see raster::Raster-class

nrows see raster::Raster-class

crs see raster::Raster-class

history see raster::Raster-class

z see raster::Raster-class

index data.frame TODO

`Extract by index` *Extract by index*

Description

`object[[i]]` can be used to extract individual objects from container classes such as `ExpVarRasterStack`, `PredictiveModelList`, `PredictionList` and `PerformanceList`.

Usage

```
## S4 method for signature DiscreteLulcRasterStack,ANY,ANY
x[[i, j, ...]]

## S4 method for signature PerformanceList,ANY,ANY
x[[i, j, ...]]

## S4 method for signature PredictionList,ANY,ANY
x[[i, j, ...]]

## S4 method for signature PredictiveModelList,ANY,ANY
x[[i, j, ...]]

## S4 method for signature PredictiveModelList,ANY,ANY
x[i, j, ..., drop = FALSE]

## S4 method for signature ContinuousLulcRasterStack,ANY,ANY
x[[i, j, ...]]
```

Arguments

`x`	An object of class DiscreteLulcRasterStack, ContinuousLulcRasterStack, PredictionList, PerformanceList, PredictiveModelList
`i`	layer number (if 'x' inherits from a RasterStack) or list index (if 'x' stores data as a list)
`j`	numeric (not used)
`...`	additional arguments (none)
`drop`	logical. If TRUE the result is coerced to the lowest possible dimension

Examples

```
## Plum Island Ecosystems

lu <- DiscreteLulcRasterStack(x=stack(pie[1:3]),
                              categories=c(1,2,3),
                              labels=c("Forest","Built","Other"),
                              t=c(0,6,14))

summary(lu[[1]])
summary(lu[[1:2]])

## Also see lulcc-package
```

FigureOfMerit *Create a FigureOfMerit object*

Description

Calculate the figure of merit at different levels and at different resolutions for a reference map at time 1, a reference map at time 2 and a simulated map at time 2.

Usage

```
FigureOfMerit(x, ...)

## S4 method for signature RasterLayer
FigureOfMerit(x, ...)

## S4 method for signature ThreeMapComparison
FigureOfMerit(x, ...)
```

Arguments

x	a ThreeMapComparison object or RasterLayer
...	additional arguments to ThreeMapComparison. Only required if x is not a ThreeMapComparison object

Details

In land use change modelling the figure of merit is the intersection of observed change and simulated change divided by the union of these, with a range of 0 (perfect disagreement) to 1 (perfect agreement). It is useful to calculate the figure of merit at three levels: (1) considering all possible transitions from all land use categories, (2) considering all transitions from specific land use categories and (3) considering a specific transition from one land use category to another.

Value

A FigureOfMerit object.

References

Pontius Jr, R.G., Peethambaram, S., Castella, J.C. (2011). Comparison of three maps at multiple resolutions: a case study of land change simulation in Cho Don District, Vietnam. Annals of the Association of American Geographers 101(1): 45-62.

See Also

plot.FigureOfMerit, ThreeMapComparison

Examples

```
## see lulcc-package examples
```

FigureOfMerit-class *Class FigureOfMerit*

Description

An S4 class for different figure of merit scores.

Slots

`tables` list of data.frames that depict the three dimensional table described by Pontius et al. (2011) at different resolutions

`factors` numeric vector of aggregation factors

`maps` list of RasterStack objects containing land use maps at different resolutions

`categories` numeric vector of land use categories

`labels` character vector corresponding to `categories`

`overall` list containing the overall figure of merit score for each aggregation factor

`category` list of numeric vectors containing category specific scores

`transition` list of matrices containing transition specific scores

getPredictiveModelInputData *Extract data to fit predictive models*

Description

Extract a data.frame containing variables required for fitting predictive models. Column names correspond to the names of lu and ef.

Usage

```
getPredictiveModelInputData(lu, ef, cells, ...)
```

Arguments

`lu`	an LulcRasterStack object
`ef`	an ExpVarRasterStack object
`cells`	index of cells to be extracted, which may be a `SpatialPoints*` object or a numeric vector representing cell numbers (see `raster::extract`)
`...`	additional arguments to as.data.frame

Value

A data.frame.

See Also

`as.data.frame`, `LulcRasterStack`, `ExpVarRasterStack`, `partition`

Examples

```
## Not run:

## Plum Island Ecosystems

lu <- DiscreteLulcRasterStack(x=stack(pie[1:3]),
                              categories=c(1,2,3),
                              labels=c("Forest","Built","Other"),
                              t=c(0,6,14))

idx <- data.frame(var=paste("ef_", formatC(1:3, width=3, flag=0)),
                  yr=rep(0,3),
                  dynamic=rep(FALSE,3))

ef <- ExpVarRasterStack(x=stack(pie[4:6]), index=idx)

part <- partition(x=lu, size=0.1, spatial=TRUE, t=0)

train.data <- getPredictiveModelInputData(lu=lu,
                                          ef=ef,
                                          cells=part[["train"]],
                                          t=0)
dim(train.data)
names(train.data)


## End(Not run)
```

LulcRasterStack *Create an LulcRasterStack object*

Description

Methods to create an LulcRasterStack object, which may be created from file or an existing Raster* object

Usage

```
DiscreteLulcRasterStack(x, ...)

## S4 method for signature Raster
DiscreteLulcRasterStack(x, ...)

## S4 method for signature RasterStack
DiscreteLulcRasterStack(x, categories, labels, t)

ContinuousLulcRasterStack(x, ...)

## S4 method for signature Raster
ContinuousLulcRasterStack(x, ...)

## S4 method for signature RasterStack
ContinuousLulcRasterStack(x, categories, labels, t)
```

Arguments

x	path (character), Raster* object or list of Raster* objects. Default behaviour is to search for files in the working directory
categories	numeric vector of land use categories in observed maps
labels	character vector (optional) with labels corresponding to categories
t	numeric vector containing the timestep of each observed map. The first timestep must be 0
...	additional arguments to raster::stack

Details

Observed land use maps should have the same extent and resolution and have the same non-NA cells. The location of non-NA cells in LulcRasterStack objects defines the region for subsequent analysis.

Value

An LulcRasterStack object.

See Also

LulcRasterStack-class, raster::stack

Examples

```
## Plum Island Ecosystems

lu <- DiscreteLulcRasterStack(x=stack(pie[1:3]),
                              categories=c(1,2,3),
                              labels=c("Forest","Built","Other"),
                              t=c(0,6,14))
lu

## Sibuyan Island
lu <- DiscreteLulcRasterStack(x=stack(sibuyan$maps[1:2]),
                              categories=c(1,2,3,4,5),
                              labels=c("forest","coconut","grass","rice","other"),
                              t=c(0,14))
```

LulcRasterStack-class *Class LulcRasterStack*

Description

A virtual S4 class for observed land use maps.

Slots

`filename` see `raster::Raster-class`
`layers` see `raster::Raster-class`
`title` see `raster::Raster-class`
`extent` see `raster::Raster-class`
`rotated` see `raster::Raster-class`
`rotation` see `raster::Raster-class`
`ncols` see `raster::Raster-class`
`nrows` see `raster::Raster-class`
`crs` see `raster::Raster-class`
`history` see `raster::Raster-class`
`z` see `raster::Raster-class`
`t` numeric vector with timesteps corresponding to each observed map
`categories` numeric vector of land use categories
`labels` character vector corresponding to `categories`

`Model-class` *Virtual class Model*

Description

A virtual S4 class to represent land use change models.

`NeighbRasterStack` *Create a NeighbRasterStack object*

Description

Methods to calculate neighbourhood values for cells in raster maps using `raster::focal`. By default the fraction of non-NA cells within the moving window (i.e. the size of the weights matrix) devoted to each land use category is calculated. This behaviour can be changed by altering the weights matrix or providing an alternative function. The resulting object can be used as the basis of neighbourhood decision rules.

Usage

```
NeighbRasterStack(x, weights, neighb, ...)

## S4 method for signature RasterLayer,list,ANY
NeighbRasterStack(x, weights, neighb,
  categories, fun = mean, ...)

## S4 method for signature RasterLayer,matrix,ANY
NeighbRasterStack(x, weights, neighb,
  categories, fun = mean, ...)

## S4 method for signature RasterLayer,ANY,NeighbRasterStack
NeighbRasterStack(x, weights,
  neighb)
```

Arguments

x	RasterLayer containing categorical data
weights	list containing a matrix of weights (the w argument in raster::focal) for each land use category. The order of list or vector elements should correspond to the order of land use categories in categories
neighb	NeighbRasterStack object. Only used if categories and weights are not provided. This option can be useful when existing NeighbRasterStack objects need to be updated because a new land use map is available, such as during the allocation procedure.
categories	numeric vector containing land use categories for which neighbourhood values should be calculated
fun	function. Input argument to focal. Default is mean
...	additional arguments to raster::focal

Value

A NeighbRasterStack object.

See Also

NeighbRasterStack-class, allowNeighb, raster::focal

Examples

```
## Plum Island Ecosystems

lu <- DiscreteLulcRasterStack(x=stack(pie[1:3]),
                              categories=c(1,2,3),
                              labels=c("Forest","Built","Other"),
                              t=c(0,6,14))

## create a NeighbRasterStack object for 1985 land use map
w1 <- matrix(data=1, nrow=3, ncol=3, byrow=TRUE)
w2 <- w1
w3 <- w1

nb1 <- NeighbRasterStack(x=lu[[1]],
                         categories=c(1,2,3),
                         weights=list(w1,w2,w3))

## update nb2 for 1991
nb2 <- NeighbRasterStack(x=lu[[2]],
                         neighb=nb1)

## plot neighbourhood map for forest
plot(nb2[[1]])
```

NeighbRasterStack-class *Class NeighbRasterStack*

Description

An S4 class for neighbourhood maps.

Slots

`filename` see `raster::Raster-class`

`layers` see `raster::Raster-class`

`title` see `raster::Raster-class`

`extent` see `raster::Raster-class`

`rotated` see `raster::Raster-class`

`rotation` see `raster::Raster-class`

`ncols` see `raster::Raster-class`

`nrows` see `raster::Raster-class`

`crs` see `raster::Raster-class`

`history` see `raster::Raster-class`

`z` see `raster::Raster-class`

`calls` list containing each call to `raster::focal`

`categories` numeric vector of land use categories for which neighbourhood maps exist

ordered *Ordered allocation*

Description

Allocate land use change using the ordered algorithm.

Usage

```
ordered(lu0, lu0.vals, tprob, nb = NULL, nb.rules = NULL,
  transition.rules = NULL, hist.vals = NULL, mask.vals = NULL, demand,
  categories, order, stochastic)
```

Arguments

`lu0`	RasterLayer showing initial land use
`lu0.vals`	numeric containing non-NA values from `lu0`
`tprob`	matrix with land use suitability values. Columns should correspond to `categories`, rows should correspond with `cells`
`nb`	neighbourhood map. See CluesModel

nb.rules	neighbourhood rules. See CluesModel documentation
transition.rules	transition rules. See CluesModel documentation
hist.vals	numeric vector detailing the number of consecutive time steps each cell has been allocated to its current land use
mask.vals	numeric vector containing binary values where 0 indicates cells that are not allowed to change
demand	matrix with demand for each land use category in terms of number of cells to be allocated. The first row should be the number of cells allocated to the initial land use map, the second row should be the number of cells to allocate in the subsequent time point
categories	numeric vector containing land use categories
order	numeric vector of land use categories in the order that change should be allocated
stochastic	Logical indicating whether or not the allocation routine should be run in stochastic mode
...	additional arguments (none)

Value

numeric vector with updated land use values.

Examples

```
## See lulcc-package examples
```

`OrderedModel` *Create a OrderedModel object*

Description

Methods to create a `OrderedModel` object to supply to `allocate`.

Usage

```
OrderedModel(observed.lulc, explanatory.variables, predictive.models, time,
  demand, history = NULL, mask = NULL, neighbourhood = NULL,
  transition.rules, neighbourhood.rules = NULL, order, ...)
```

Arguments

observed.lulc	an LulcRasterStack
explanatory.variables	an ExpVarRasterStack object
predictive.models	a PredictiveModelList object
time	numeric vector containing timesteps over which simulation will occur

demand	matrix with demand for each land use category in terms of number of cells to be allocated. The first row should be the number of cells allocated to the initial observed land use map (i.e. the land use map for time 0)
history	RasterLayer containing land use history (values represent the number of years the cell has contained the current land use category)
mask	RasterLayer containing binary values where 0 indicates cells that are not allowed to change
neighbourhood	an object of class NeighbRasterStack
transition.rules	matrix with land use change decision rules
neighbourhood.rules	numeric with neighbourhood decision rules
order	numeric vector of land use categories in the order that change should be allocated
...	additional arguments (none)

Value

A OrderedModel object.

References

Fuchs, R., Herold, M., Verburg, P.H., and Clevers, J.G.P.W. (2013). A high-resolution and harmonized model approach for reconstructing and analysing historic land changes in Europe, Biogeosciences, 10:1543-1559.

See Also

OrderedModel-class, allocate

Examples

```
## see lulcc-package examples
```

OrderedModel-class *Class OrderedModel*

Description

An S4 class to represent inputs to the Ordered allocation procedure

Slots

observed.lulc an LulcRasterStack object

explanatory.variables an ExpVarRasterStack object

predictive.models a PredictiveModelList object

time numeric vector of timesteps over which simulation will occur

demand matrix containing demand scenario

history RasterLayer showing land use history or NULL

`mask` RasterLayer showing masked areas or NULL

`neighbourhood` NeighbRasterStack object or NULL

`transition.rules` matrix with land use change decision rules

`neighbourhood.rules` numeric with neighbourhood decision rules

`order` numeric vector of land use categories in the order that change should be allocated

`categories` numeric vector of land use categories

`labels` character vector corresponding to `categories`

`partition` *Partition raster data*

Description

Divide a categorical raster map into training and testing partitions. A wrapper function for `caret::createDataPartition` (Kuhn, 2008) to divide a categorical raster map into training and testing partitions.

Usage

```
partition(x, ...)

## S4 method for signature RasterLayer
partition(x, size = 0.5, spatial = TRUE, ...)

## S4 method for signature DiscreteLulcRasterStack
partition(x, size = 0.5, spatial = TRUE,
  t, ...)

## S4 method for signature ContinuousLulcRasterStack
partition(x, size = 0.5,
  spatial = TRUE, ...)
```

Arguments

`x`	RasterLayer with categorical data
`size`	numeric value between zero and one indicating the proportion of non-NA cells that should be included in the training partition. Default is 0.5, which results in equally sized partitions
`spatial`	logical. If TRUE, the function returns a SpatialPoints object with the coordinates of cells in each partition. If FALSE, the cell numbers are returned
`t`	numeric corresponding to one of the time points for which a land use map is available.
`...`	additional arguments (none)

Value

A list containing the following components:

`train` a SpatialPoints object or numeric vector indicating the cells in the training partition

`test` a SpatialPoints object or numeric vector indicating the cells in the testing partition

`all` a SpatialPoints object or numeric vector indicating all non-NA cells in the study region

References

Kuhn, M. (2008). Building predictive models in R using the caret package. Journal of Statistical Software, 28(5), 1-26.

See Also

caret::createDataPartition

Examples

```
## Not run:

## Plum Island Ecosystems

lu <- DiscreteLulcRasterStack(x=stack(pie[1:3]),
                              categories=c(1,2,3),
                              labels=c("Forest","Built","Other"),
                              t=c(0,6,14))

part <- partition(x=lu, size=0.05, spatial=TRUE, t=0)

plot(lu[[1]])
points(part[["train"]])


## End(Not run)
```

PerformanceList *Create a PerformanceList object*

Description

This function uses different measures to evaluate multiple ROCR::prediction objects stored in a PredictionList object.

Usage

```
PerformanceList(pred, measure, x.measure = "cutoff", ...)
```

Arguments

pred	an object of class PredictionList
measure	performance measure to use for the evaluation. See ROCR::performance
x.measure	a second performance measure. See ROCR::performance
...	additional arguments to ROCR::performance

Value

A PerformanceList object.

References

Sing, T., Sander, O., Beerenwinkel, N., Lengauer, T. (2005). ROCR: visualizing classifier performance in R. Bioinformatics 21(20):3940-3941.

See Also

performance, PredictionList

Examples

```
## see lulcc-package examples
```

PerformanceList-class *Class PerformanceList*

Description

An S4 class that extends ROCR::performance-class to hold the results of multiple model evaluations.

Slots

performance list of ROCR performance objects. Each object is calculated for the corresponding ROCR prediction object held in the PredictionList object supplied to the constructor function

auc numeric vector containing the area under the curve for each performance object

categories numeric vector of land use categories for which performance objects were created

labels character vector with labels corresponding to categories

---

pie *Land use change dataset for Plum Island Ecosystem*

Description

Dataset containing land use maps for 1985, 1991 and 1999 and several explanatory variables derived from Pontius and Parmentier (2014).

Usage

```
pie
```

Format

A list containing the following elements:

lu_pie_1985 RasterLayer showing land use in 1985 (forest, built, other)

lu_pie_1991 RasterLayer showing land use in 1991

lu_pie_1999 RasterLayer showing land use in 1999

ef_001 RasterLayer showing elevation

ef_002 RasterLayer showing slope

ef_003 RasterLayer showing distance to built land in 1985

References

Pontius Jr, R. G., & Parmentier, B. (2014). Recommendations for using the relative operating characteristic (ROC). Landscape ecology, 29(3), 367-382.

Examples

```
data(pie)
```

`plot` *Plot method for objects based on Raster* data*

Description

Plot `lulcc` objects based on Raster* data

Usage

```
## S3 method for class ContinuousLulcRasterStack
plot(x, y, ...)

## S3 method for class DiscreteLulcRasterStack
plot(x, y, ...)

## S3 method for class ThreeMapComparison
plot(x, y, category, factors, ...)

## S4 method for signature ContinuousLulcRasterStack,ANY
plot(x, y, ...)

## S4 method for signature DiscreteLulcRasterStack,ANY
plot(x, y, ...)

## S4 method for signature ThreeMapComparison,ANY
plot(x, y, category, factors, ...)
```

Arguments

`x`	an object from `lulcc` containing Raster data
`y`	not used
`category`	numeric
`factors`	numeric
`...`	additional arguments to `rasterVis::levelplot`

Value

A trellis object.

See Also

`rasterVis::levelplot`

Examples

```
## see lulcc-package examples
```

`plot.AgreementBudget` *Plot method for AgreementBudget objects*

Description

Plot an `AgreementBudget` object.

Usage

```
## S3 method for class AgreementBudget
plot(x, y, from, to,
  col = RColorBrewer::brewer.pal(5, "Set2"), key, scales, xlab, ylab, ...)

## S4 method for signature AgreementBudget,ANY
plot(x, y, from, to,
  col = RColorBrewer::brewer.pal(5, "Set2"), key, scales, xlab, ylab, ...)
```

Arguments

`x`	an AgreementBudget object
`y`	not used
`from`	optional numeric value representing a land use category. If provided without `to` the figure of merit for all transitions from this category will be plotted
`to`	similar to `from`. If provided with a valid `from` argument the transition defined by these two arguments (i.e. `from` -> `to`) will be plotted
`col`	character specifying the plotting colour. Default is to use the 'Set2' palette from `RColorBrewer`
`key`	list. See `lattice::xyplot`
`scales`	list. See `lattice::xyplot`
`xlab`	character or expression. See `lattice::xyplot`
`ylab`	character or expression. See `lattice::xyplot`
`...`	additional arguments to `lattice::xyplot`

Details

The plot layout is based on work presented in Pontius et al. (2011)

Value

A trellis object.

References

Pontius Jr, R.G., Peethambaram, S., Castella, J.C. (2011). Comparison of three maps at multiple resolutions: a case study of land change simulation in Cho Don District, Vietnam. Annals of the Association of American Geographers 101(1): 45-62.

See Also

AgreementBudget, lattice::xyplot

Examples

```
## see lulcc-package examples
```

---

plot.FigureOfMerit *Plot method for FigureOfMerit objects*

Description

Plot the overall, category-specific or transition-specific figure of merit at different resolutions.

Usage

```
## S3 method for class FigureOfMerit
plot(x, y, ..., from, to,
  col = RColorBrewer::brewer.pal(8, "Set2"), type = "b", key, scales, xlab,
  ylab)

## S4 method for signature FigureOfMerit,ANY
plot(x, y, ..., from, to,
  col = RColorBrewer::brewer.pal(8, "Set2"), type = "b", key, scales, xlab,
  ylab)
```

Arguments

x	a FigureOfMerit object
y	not used
from	optional numeric value representing a land use category. If provided without to the figure of merit for all transitions from this category will be plotted
to	similar to from. If provided with a valid from argument the transition defined by these two arguments (i.e. from -> to) will be plotted. It is possible to include more than one category in which case the different transitions will be included on the same plot
col	character specifying the plotting colour. Default is to use the 'Set2' palette from RColorBrewer
type	character. See lattice::panel.xyplot
key	list. See lattice::xyplot
scales	list. See lattice::xyplot
xlab	character or expression. See lattice::xyplot
ylab	character or expression. See lattice::xyplot
...	additional arguments to lattice::xyplot

Value

A trellis object.

See Also

FigureOfMerit, lattice::xyplot, lattice::panel.xyplot

Examples

```
## see lulcc-package examples
```

---

plot.PerformanceList *Plot method for PerformanceList objects*

Description

Plot the the ROC curve for each performance object in a PerformanceList object. If more than one PerformanceList objects are provided ROC curves for the same land use category from different objects are included on the same plot for model comparison.

Usage

```
## S3 method for class PerformanceList
plot(x, y, multipanel = TRUE, type = "l",
  abline = list(c(0, 1), col = "grey"), col = RColorBrewer::brewer.pal(9,
  "Set1"), key.args = NULL, ...)

## S4 method for signature list,ANY
plot(x, y, multipanel = TRUE, type = "l",
  abline = list(c(0, 1), col = "grey"), col = RColorBrewer::brewer.pal(9,
  "Set1"), key.args = NULL, ...)
```

Arguments

x	either a single PerformanceList object or a list of these. If a list is provided it must be named.
y	not used
multipanel	logical. If TRUE, create a trellis plot where the number of panels equals the number of PerformanceList objects. Otherwise, create a single plot for each PerformanceList object
type	character. See lattice::panel.xyplot
abline	list. See lattice::panel.xyplot
col	character. Plotting colour
key.args	list containing additional components to be passed to the key argument of lattice::xyplot
...	additional arguments to lattice::xyplot

Value

A trellis object.

See Also

PerformanceList, lattice::xyplot

Examples

```
## see lulcc-package examples
```

`predict.PredictiveModelList` *Predict allocation suitability*

Description

Estimate allocation suitability with predictive models.

Usage

```
## S3 method for class PredictiveModelList
predict(object, newdata, data.frame = FALSE,
  ...)
```

Arguments

`object`	a PredictiveModelList object
`newdata`	data.frame containing new data
`data.frame`	logical indicating whether the function should return a matrix (default) or data.frame
`...`	additional arguments to `predict` methods

Details

This function is usually called from `allocate` to calculate land use suitability at each timestep. However, it may also be used to produce suitability maps (see examples).

Value

A matrix.

See Also

`predict`, `allocate`

Examples

```
## Not run:

## Plum Island Ecosystems

lu <- DiscreteLulcRasterStack(x=stack(pie[1:3]),
                              categories=c(1,2,3),
                              labels=c("Forest","Built","Other"),
                              t=c(0,6,14))

idx <- data.frame(var=c("ef_001","ef_002","ef_003"),
                  yr=c(0,0,0),
                  dynamic=c(FALSE,FALSE,FALSE))
ef <- ExpVarRasterStack(x=stack(pie[4:6]), index=idx)
```

```
part <- partition(x=lu, size=0.1, spatial=TRUE, t=0)
train.data <- getPredictiveModelInputData(lu=lu,
                                          ef=ef,
                                          cells=part[["train"]],
                                          t=0)

forest.form <- as.formula("Forest ~ ef_001 + ef_002")
built.form <- as.formula("Built ~ ef_001 + ef_002 + ef_003")
other.form <- as.formula("Other ~ ef_001 + ef_002")

forest.glm <- glm(forest.form, family=binomial, data=train.data)
built.glm <- glm(built.form, family=binomial, data=train.data)
other.glm <- glm(other.form, family=binomial, data=train.data)
glm.mods <- PredictiveModelList(list(forest.glm, built.glm, other.glm),
                                categories=lu@categories,
                                labels=lu@labels)

all.data <- as.data.frame(x=ef, cells=part[["all"]])
probmaps <- predict(object=glm.mods,
                    newdata=all.data,
                    data.frame=TRUE)

points <- rasterToPoints(lu[[1]], spatial=TRUE)
probmaps <- SpatialPointsDataFrame(points, probmaps)
probmaps <- rasterize(x=probmaps, y=lu[[1]],
                      field=names(probmaps))
plot(probmaps)


## End(Not run)
```

`PredictionList` *Create a PredictionList object*

Description

This function creates a `ROCR::prediction` object for each predictive model in a `PredictiveModelList` object. It should be used with `PerformanceList` to evaluate multiple models with exactly the same criteria while keeping track of which model corresponds to which land use category.

Usage

```
PredictionList(models, newdata, ...)
```

Arguments

`models`	a PredictiveModelList object
`newdata`	a data.frame containing new data
`...`	additional arguments to `ROCR::prediction`

Value

A `PredictionList` object.

References

Sing, T., Sander, O., Beerenwinkel, N., Lengauer, T. (2005). ROCR: visualizing classifier performance in R. Bioinformatics 21(20):3940-3941.

See Also

link{PerformanceList}, ROCR::prediction

Examples

```
## see lulcc-package examples
```

PredictionList-class *Class PredictionList*

Description

An S4 class that extends ROCR::prediction-class to hold the results of multiple model predictions.

Slots

`prediction` a list of ROCR::prediction-class objects. These objects are calculated for each statistical model in the `PredictiveModelList` object supplied to the constructor function

`categories` numeric vector of land use categories for which `prediction` objects were created

`labels` character vector with labels corresponding to `categories`

PredictiveModelList *Create PredictiveModelList object*

Description

Crete an object of class PredictiveModelList.

Usage

```
PredictiveModelList(models, categories, labels, ...)
```

Arguments

`models`	list containing predictive models
`categories`	numeric vector of land use categories in observed maps
`labels`	character vector with labels corresponding to `categories`
`...`	additional arguments (none)

Value

A PredictiveModelList object

Examples

```
## see lulcc-package examples
```

PredictiveModelList-class
Class PredictiveModelList

Description

An S4 class to hold multiple mathematical models for different land use categories belonging to the same map.

Slots

`models` list of predictive models

`categories` numeric vector of land use categories

`labels` character vector with labels corresponding to `categories`

resample,ExpVarRasterStack,Raster-method
Resample maps in ExpVarRasterStack object or list

Description

A wrapper function for `raster::resample` to resample raster objects in an ExpVarRasterStack object or list.

Usage

```
## S4 method for signature ExpVarRasterStack,Raster
resample(x, y, method = "ngb", ...)

## S4 method for signature list,Raster
resample(x, y, method = "ngb", ...)
```

Arguments

`x`	an ExpVarRasterStack object or list of Raster* maps to be resampled
`y`	Raster* object with parameters that `x` should be resampled to
`method`	method used to compute values for the new RasterLayer, should be `"bilinear"` for bilinear interpolation, or `"ngb"` for nearest neighbour
`...`	additional arguments to `raster::resample`

Value

An ExpVarRasterStack object or list, depending on `x`.

See Also

`ExpVarRasterStack`, `raster::resample`

Examples

```
## see lulcc-examples
```

roundSum *Round elements in matrix or data.frame rows*

Description

Round all numbers in a matrix or data.frame while ensuring that all rows sum to the same value.

Usage

```
roundSum(x, n, digits = 0, ...)
```

Arguments

x	matrix or data.frame
n	numeric specifying the target sum for each row in x
digits	integer indicating the number of decimal places to be used
...	additional arguments (none)

Details

The main application of roundSum is to ensure that each row in the demand matrix specifies exactly the number of cells to be allocated to each land use category for the respective timestep. It may also be used to convert the units of demand to number of cells.

Value

A matrix.

Examples

```
## Not run:

## Sibuyan Island

## load observed land use data and create demand scenario
obs <- LulcRasterStack(x=sibuyan$maps,
                   pattern="lu",
                   categories=c(1,2,3,4,5),
                   labels=c("Forest","Coconut","Grass","Rice","Other"),
                   t=c(0,14))

dmd <- approxExtrapDemand(obs, tout=0:14)
apply(dmd, 1, sum)

## artificially perturb for illustration purposes
dmd <- dmd * runif(1)
apply(dmd, 1, sum)

## use roundSum to correct demand scenario
ncell <- length(which(!is.na(getValues(sibuyan$maps$lu_sib_1997))))
ncell
dmd <- roundSum(dmd, ncell=ncell)
```

```
apply(dmd, 1, sum)


## End(Not run)
```

`show,PredictiveModelList-method` *Show objects*

Description

Show lulcc objects

Usage

```
## S4 method for signature PredictiveModelList
show(object)

## S4 method for signature PredictionList
show(object)

## S4 method for signature PerformanceList
show(object)

## S4 method for signature Model
show(object)

## S4 method for signature ThreeMapComparison
show(object)
```

Arguments

`object`	an object belonging to one of the classes in `lulcc`

Value

Null

`sibuyan` *Land use change dataset for Sibuyan Island*

Description

Dataset containing land use map for 1997 and several explanatory variables for Sibuyan Island derived from Verburg et al. (2002). Data are modified by Peter Verburg to demonstrate the CLUE-s model; as such the dataset should not be used for purposes other than demonstration.

Usage

```
sibuyan
```

Format

A list containing the following components:

maps list containing the following RasterLayers:

- **lu_sib_1997** RasterLayer with land use in 1997 (forest, coconut, grassland, rice, other)
- **ef_001** RasterLayer showing distance to sea
- **ef_002** RasterLayer showing mean population density
- **ef_003** RasterLayer showing occurrence of diorite rock
- **ef_004** RasterLayer showing occurrence of ultramafic rock
- **ef_005** RasterLayer showing occurrence of sediments
- **ef_006** RasterLayer showing areas with no erosion
- **ef_007** RasterLayer showing areas with moderate erosion
- **ef_008** RasterLayer showing elevation
- **ef_009** RasterLayer showing slope
- **ef_010** RasterLayer showing aspect
- **ef_011** RasterLayer showing distance to roads in 1997
- **ef_012** RasterLayer showing distance to urban areas in 1997
- **ef_013** RasterLayer showing distance to streams
- **restr1** RasterLayer showing location of current national park
- **restr2** RasterLayer showing location of proposed national park

demand list of matrices with different demand scenarios:

- **demand1** data.frame with demand scenario representing slow growth scenario
- **demand2** data.frame with demand scenario representing fast growth scenario
- **demand3** data.frame with demand scenario representing land use change primarily for food production

References

Verburg, P.H., Soepboer, W., Veldkamp, A., Limpiada, R., Espaldon, V., Mastura, S.S (2002). Modeling the Spatial Dynamics of Regional Land Use: The CLUE-S Model. Environmental Management 30(3): 391-405.

Examples

```
data(sibuyan)
```

---

subset,PredictiveModelList-method

Subset objects

Description

Extract a subset of objects from container classes such as `ExpVarRasterStack`, `PredictiveModelList`, `PredictionList` and `PerformanceList`.

Usage

```
## S4 method for signature PredictiveModelList
subset(x, subset, drop = FALSE, ...)

## S4 method for signature PerformanceList
subset(x, subset, ...)

## S4 method for signature PredictionList
subset(x, subset, ...)
```

Arguments

x	an object of class ExpVarRasterStack, PredictiveModelList, PredictionList or PerformanceList
subset	integer or character indicating the objects to be extracted
drop	logical
...	additional arguments (none)

Value

Subsetted object, possibly simplified

Examples

```
## Sibuyan Island

lu <- DiscreteLulcRasterStack(x=stack(sibuyan$maps[1:2]),
                              categories=c(1,2,3,4,5),
                              labels=c("forest","coconut","grass","rice","other"),
                              t=c(0,14))


summary(lu)
lu <- subset(lu, subset=names(lu)[1])
summary(lu)

## load explanatory variables
idx <- data.frame(var=paste("ef_", formatC(1:13, width=3, flag=0)),
                  yr=rep(0,13),
                  dynamic=rep(FALSE,13))

ef <- ExpVarRasterStack(x=stack(sibuyan$maps[3:15]), index=idx)

summary(ef)
ef <- subset(ef, subset=1:5)
summary(ef)
```

summary *Summary*

Description

Summarise lulcc objects

Usage

```
summary(object, ...)

## S4 method for signature LulcRasterStack
summary(object, ...)

## S4 method for signature ExpVarRasterStack
summary(object, ...)

## S4 method for signature Model
summary(object, ...)
```

Arguments

object	an object belonging to one of the classes in lulcc
...	additional arguments (none)

Value

A matrix, data.frame or list

ThreeMapComparison *Evaluate allocation performance with three maps*

Description

An implementation of the method described by Pontius et al. (2011), which compares a reference map at time 1, a reference map at time 2 and a simulated map at time 2 to evaluate allocation performance at multiple resolutions while taking into account persistence. The method quantifies disagreement within coarse squares (minor allocation disagreement), disagreement between coarse squares (major allocation disagreement), disagreement about the quantity of land use change and agreement.

Usage

```
ThreeMapComparison(x, x1, y1, ...)

## S4 method for signature RasterLayer,RasterLayer,RasterLayer
ThreeMapComparison(x, x1, y1,
  factors, categories, labels, ...)
```

Arguments

x	either a RasterLayer of observed land use at time 0 or an object inheriting from class Model
x1	a RasterLayer of observed land use at a subsequent time. Only required if x is also a RasterLayer
y1	a RasterLayer of simulated land use corresponding to x1. Only required if x is also a RasterLayer
factors	numeric vector of aggregation factors (equivalent to the 'fact' argument to raster::aggregate representing the resolutions at which model performance should be tested
categories	numeric vector of land use categories in observed maps. Only required if x is a RasterLayer
labels	character vector (optional) with labels corresponding to categories. Only required if x is a RasterLayer
...	additional arguments to raster::aggregate

Value

A ThreeMapComparison object.

References

Pontius Jr, R.G., Peethambaram, S., Castella, J.C. (2011). Comparison of three maps at multiple resol utions: a case study of land change simulation in Cho Don District, Vietnam. Annals of the Association of American Geographers 101(1): 45-62.

See Also

AgreementBudget, FigureOfMerit, raster::aggregate

Examples

```
## see lulcc-package examples
```

ThreeMapComparison-class

Class ThreeMapComparison

Description

An S4 class to hold results of a comparison between a reference map for time 1, a reference map for time 2 and a simulation map for time 2 using the the method described by Pontius et al. (2011).

Slots

tables list of data.frames that depict the three dimensional table described by Pontius et al. (2011) at different resolutions

factors numeric vector of aggregation factors

maps list of RasterStack objects containing land use maps at different resolutions

categories numeric vector of land use categories

labels character vector corresponding to categories

References

Pontius Jr, R.G., Peethambaram, S., Castella, J.C. (2011). Comparison of three maps at multiple resol utions: a case study of land change simulation in Cho Don District, Vietnam. Annals of the Association of American Geographers 101(1): 45-62.

`total` *Total number of cells in a Raster* object*

Description

Count the area or number of cells belonging to each category in a Raster* object.

Usage

```
total(x, ...)

## S4 method for signature DiscreteLulcRasterStack
total(x, categories, ...)

## S4 method for signature ContinuousLulcRasterStack
total(x, categories, ...)

## S4 method for signature Raster
total(x, categories, ...)
```

Arguments

`x`	Raster* object
`categories`	numeric vector containing land use categories. Only cells belonging to these categories will be counted
`...`	additional arguments (none)

Details

If x is a DiscreteLulcRasterStack object this function returns the number of cells belonging to each category. If x is a ContinuousLulcRasterStack object the function returns the sum of the fractions of the various land use categories.

Value

A list containing the following components:

`total` a matrix containing the total number of cells belonging to each category. Rows represent layers in the input Raster* object

`categories` the categories included in the calculation

Examples

```
## Sibuyan Island

lu <- DiscreteLulcRasterStack(x=stack(sibuyan$maps[1:2]),
                              categories=c(1,2,3,4,5),
                              labels=c("forest","coconut","grass","rice","other"),
                              t=c(0,14))

total(x=lu)
total(x=lu[[1]])
total(x=lu[[2]])
```

`updateDataFrame` *Update data frame*

Description

Function to update a data frame holding model variables with values of dynamic covariables for a new time point. This function is used internally by allocation routines.

Usage

```
updateDataFrame(x, ...)

## S4 method for signature ExpVarRasterStack
updateDataFrame(x, y, cells, time, ...)
```

Arguments

`x`	ExpVarRasterStack object
`y`	data.frame to update
`cells`	index of cells to be extracted and added to the data frame (see `as.data.frame`)
`time`	numeric indicating the time for which the data frame should be updated
`...`	additional arguments (none)

Value

data.frame

Index

∗Topic **datasets**
pie, 41
sibuyan, 51
[,PredictiveModelList,ANY,ANY-method *(Extract by index)*, 29
[[,ContinuousLulcRasterStack,ANY,ANY-method *(Extract by index)*, 29
[[,DiscreteLulcRasterStack,ANY,ANY-method *(Extract by index)*, 29
[[,PerformanceList,ANY,ANY-method *(Extract by index)*, 29
[[,PredictionList,ANY,ANY-method *(Extract by index)*, 29
[[,PredictiveModelList,ANY,ANY-method *(Extract by index)*, 29

aggregate, *55*
AgreementBudget, 8, *43, 44*, *55*
AgreementBudget,RasterLayer-method *(AgreementBudget)*, 8
AgreementBudget,ThreeMapComparison-method *(AgreementBudget)*, 8
AgreementBudget-class, 9
allocate, 9, *11*, *18, 19*, *21, 22*, *37, 38*, *46*
allocate,ClueModel-method *(allocate)*, 9
allocate,CluesModel-method *(allocate)*, 9
allocate,OrderedModel-method *(allocate)*, 9
allow, 10, *13*
allowNeighb, *11*, 12, *35*
approxExtrap, *14*
approxExtrapDemand, 13
approxExtrapDemand,DiscreteLulcRasterStack-method *(approxExtrapDemand)*, 13
approxExtrapDemand,LulcRasterStack-method *(approxExtrapDemand)*, 13
as.data.frame, *16*, *31*, *57*
as.data.frame,ContinuousLulcRasterStack-method *(as.data.frame.ExpVarRasterStack)*, 15
as.data.frame,DiscreteLulcRasterStack-method *(as.data.frame.ExpVarRasterStack)*, 15
as.data.frame,ExpVarRasterStack-method *(as.data.frame.ExpVarRasterStack)*, 15
as.data.frame.ContinuousLulcRasterStack *(as.data.frame.ExpVarRasterStack)*, 15
as.data.frame.DiscreteLulcRasterStack *(as.data.frame.ExpVarRasterStack)*, 15
as.data.frame.ExpVarRasterStack, 15

c.PredictiveModelList, 16
clue, 17
ClueModel, 18
ClueModel-class, 20
clues, 20
CluesModel, *10*, 21
CluesModel-class, 23
compareAUC, 23
compareAUC,list-method *(compareAUC)*, 23
compareAUC,PredictionList-method *(compareAUC)*, 23
ContinuousLulcRasterStack *(LulcRasterStack)*, 32
ContinuousLulcRasterStack,Raster-method *(LulcRasterStack)*, 32
ContinuousLulcRasterStack,RasterStack-method *(LulcRasterStack)*, 32
ContinuousLulcRasterStack-class, 24
createDataPartition, *39, 40*
crosstab, *25, 26*
crossTabulate, 25
crossTabulate,DiscreteLulcRasterStack,ANY-method *(crossTabulate)*, 25
crossTabulate,RasterLayer,RasterLayer-method *(crossTabulate)*, 25

DiscreteLulcRasterStack *(LulcRasterStack)*, 32
DiscreteLulcRasterStack,Raster-method *(LulcRasterStack)*, 32
DiscreteLulcRasterStack,RasterStack-method *(LulcRasterStack)*, 32
DiscreteLulcRasterStack-class, 26

ExpVarRasterStack, *15, 16*, 27, *31*, *49*
ExpVarRasterStack,character-method (ExpVarRasterStack), 27
ExpVarRasterStack,RasterStack-method (ExpVarRasterStack), 27
ExpVarRasterStack-class, 28
extract, *15*, *31*
Extract by index, 29

FigureOfMerit, *8*, 30, *45*, *55*
FigureOfMerit,RasterLayer-method (FigureOfMerit), 30
FigureOfMerit,ThreeMapComparison-method (FigureOfMerit), 30
FigureOfMerit-class, 31
focal, *34–36*

getPredictiveModelInputData, 31

layerize, *15*
levelplot, *42*
lulcc-package, 3
LulcRasterStack, *15, 16*, *26*, *31*, 32
LulcRasterStack-class, 33

Model-class, 34

NeighbRasterStack, *13*, 34
NeighbRasterStack,RasterLayer,ANY,NeighbRasterStack-method (NeighbRasterStack), 34
NeighbRasterStack,RasterLayer,list,ANY-method (NeighbRasterStack), 34
NeighbRasterStack,RasterLayer,matrix,ANY-method (NeighbRasterStack), 34
NeighbRasterStack-class, 36

ordered, 36
OrderedModel, 37
OrderedModel-class, 38

panel.xyplot, *44, 45*
partition, *16*, *31*, 39
partition,ContinuousLulcRasterStack-method (partition), 39
partition,DiscreteLulcRasterStack-method (partition), 39
partition,RasterLayer-method (partition), 39
performance, *24*, *40, 41*
PerformanceList, 40, *45*, *47*
PerformanceList-class, 41
pie, 41
plot, 42
plot,AgreementBudget,ANY-method (plot.AgreementBudget), 43
plot,ContinuousLulcRasterStack,ANY-method (plot), 42
plot,DiscreteLulcRasterStack,ANY-method (plot), 42
plot,FigureOfMerit,ANY-method (plot.FigureOfMerit), 44
plot,list,ANY-method (plot.PerformanceList), 45
plot,ThreeMapComparison,ANY-method (plot), 42
plot.AgreementBudget, *8*, 43
plot.ContinuousLulcRasterStack (plot), 42
plot.DiscreteLulcRasterStack (plot), 42
plot.FigureOfMerit, *30*, 44
plot.PerformanceList, 45
plot.ThreeMapComparison (plot), 42
predict, *46*
predict.PredictiveModelList, 46
prediction, *23*, *40*, *47, 48*
PredictionList, *23, 24*, *40, 41*, 47
PredictionList-class, 48
PredictiveModelList, 48
PredictiveModelList-class, 49

resample, *49*
resample,ExpVarRasterStack,Raster-method, 49
resample,list,Raster-method (resample,ExpVarRasterStack,Raster-method), 49
roundSum, 50

show,Model-method (show,PredictiveModelList-method), 51
show,PerformanceList-method (show,PredictiveModelList-method), 51
show,PredictionList-method (show,PredictiveModelList-method), 51
show,PredictiveModelList-method, 51
show,ThreeMapComparison-method (show,PredictiveModelList-method), 51
sibuyan, 51
stack, *27, 28*, *33*
subset,PerformanceList-method (subset,PredictiveModelList-method), 52

subset,PredictionList-method
(subset,PredictiveModelList-method), 52
subset,PredictiveModelList-method, 52
summary, 54
summary,ExpVarRasterStack-method
(summary), 54
summary,LulcRasterStack-method
(summary), 54
summary,Model-method (summary), 54

ThreeMapComparison, *8*, *30*, 54
ThreeMapComparison,RasterLayer,RasterLayer,RasterLayer-method
(ThreeMapComparison), 54
ThreeMapComparison-class, 55
total, 56
total,ContinuousLulcRasterStack-method
(total), 56
total,DiscreteLulcRasterStack-method
(total), 56
total,Raster-method (total), 56

updateDataFrame, 57
updateDataFrame,ExpVarRasterStack-method
(updateDataFrame), 57

xyplot, *43–45*

B Manual for *Hydro*

Package 'Hydro'

July 28, 2016

Type Package

Title Classes and methods for hydrological analysis

Version 0.0.1

Date 2015-02-25

Author Simon Moulds, Wouter Buytaert, Dominik Reusser, Jon Olav Skoien, Edzer Pebesma and Claudia Vitolo

Maintainer Simon Moulds <simon.moulds10@imperial.ac.uk>, Wouter Buytaert <w.buytaert@imperial.ac.uk>

Description Classes and methods for hydrological analysis.

Depends methods,
spacetime,
raster,
sp,
R (>= 3.1.0),

License GPL (>= 2)

LazyData true

Imports zoo,
xts,
gstat,
automap,
topmodel

Suggests RObsDat,
ncdf,
testthat,
knitr

Collate 'Class-xts.R'
'Class-HydroSTF.R'
'Class-HydroSTS.R'
'Class-HydroSTI.R'
'Arith.R'
'Class-HydroCatchment.R'
'Hydro-package.R'
'HydroCatchment-methods.R'
'HydroSTF-methods.R'
'HydroSTI-methods.R'
'HydroSTS-methods.R'

'Math.R'
'aggregate.R'
'as.spacetime.R'
'c.HydroCatchment.R'
'coerce.R'
'compare.R'
'crop.R'
'crs.R'
'data.R'
'diff.R'
'dim.R'
'extent.R'
'extract.R'
'geometry.R'
'getDataObject.R'
'gridded.R'
'index.R'
'metadata.R'
'na.R'
'names.R'
'over.R'
'resample.R'
'show.R'
'spTransform.R'
'units.R'
'window.R'

VignetteBuilder knitr

R topics documented:

Hydro-package . . . 3
aggregate . . . 3
Arith . . . 5
as.spacetime . . . 7
compare . . . 8
crop . . . 10
crs . . . 11
diff . . . 11
dim . . . 12
extent . . . 13
extract . . . 14
Extract by index . . . 15
geometry . . . 16
getDataObject . . . 17
gridded . . . 18
hc . . . 19
HydroCatchment-class . . . 19
HydroSTF-class . . . 21
HydroSTF.array-class . . . 23
HydroSTF.raster-class . . . 24
HydroSTFDF-class . . . 25

HydroSTI-class . . . 26
HydroSTIDF-class . . . 27
HydroSTS-class . . . 28
HydroSTSDF-class . . . 29
interval . . . 30
Math . . . 31
metadata . . . 32
names . . . 34
over . . . 35
plyn_aws_data . . . 36
plyn_flow_data . . . 37
regular . . . 37
resample . . . 38
spTransform . . . 39
units . . . 40
window . . . 40

Index **42**

`Hydro-package` *Hydro: an R package for hydrological analysis*

Description

Hydro provides a set of classes and methods to facilitate hydrological data analysis and workflow orchestration within the R environment.

Author(s)

Simon Moulds <simon.moulds10@imperial.ac.uk>

`aggregate` *Spatio-temporal aggregation*

Description

Spatio-temporal aggregation for HydroST objects. Methods for HydroST*DF, objects extend those defined in package spacetime. Methods for gridded HydroSTF objects additionally allow spatial aggregation using methods defined in package raster.

Usage

```
## S3 method for class HydroSTFDF
aggregate(x, by, FUN, ..., simplify = FALSE, md)

## S3 method for class HydroSTIDF
aggregate(x, by, FUN, ..., simplify = FALSE, md)

## S3 method for class HydroSTSDF
aggregate(x, by, FUN, ..., simplify = FALSE, md)
```

```
## S3 method for class HydroSTF.array
aggregate(x, by, FUN, ..., simplify = FALSE, md)

## S3 method for class HydroSTF.raster
aggregate(x, by, FUN, ..., simplify = FALSE, md,
  fact)
```

Arguments

x	HydroST object to be aggregated in space or time.
by	geometry over which attributes in x are aggregated (this can be a Spatial geometry or an ST geometry), or temporal aggregation, such as "month", "10 minutes", or a function such as as.yearmon; see aggregate.zoo. In case x is of class STFDF, argument by may be "time" or "space", in which cases aggregation over all time or all space is carried out.
FUN	aggregation function
simplify	boolean; if TRUE, and space or time dimensions can be dropped, the simpler (Spatial or xts) object will be returned
md	list containing metadata entries with which to update the metadata of x. In particular, users may wish to update DataType entry if an aggregation function such as min or max is used
fact	integer. Aggregation factor expressed as number of cells in each direction (horizontally and vertically). Or two integers (horizontal and vertical aggregation factor) or three integers (when also aggregating over layers). See raster::aggregate
...	arguments passed on to function FUN

Details

Aggregating an object in space and time changes its data type (as defined in CUAHSI's DataTypeCV table) in ways that are difficult to predict. For this reason the DataType of the aggregated HydroST object is set to "Unknown" and a warning message reminds the user to update DataType accordingly.

Value

An object of class x aggregated over the geometry of by or, if simplify is TRUE and space or time can be dropped, an object of class xts or Spatial

See Also

over::over, raster::aggregate

Examples

```
pts <- data.frame(x = 1:5, y = 1:5)
coordinates(pts) <- ~x+y
ts  <- seq(as.POSIXct("2000-01-01", tz="GMT"), by="1 hour", length.out=168)
dat <- data.frame(data=runif(840))
x   <- HydroSTF(STFDF(sp=pts, time=ts, data=dat))

## aggregate by time
x.sp    <- aggregate(x, by="time", FUN=sum, na.rm=TRUE)
```

```
summary(x.sp)

## aggregate by space
x.ts    <- aggregate(x, by="space", FUN=mean, na.rm=TRUE)
plot(x.ts, ylab=units(x))

## aggregate to daily time step
x.daily <- aggregate(x, by="1 day", FUN=sum, na.rm=TRUE)
interval(x.daily, units="days")

## aggregate to daily using HydroST object
aggregate(x, by=x.daily, FUN=sum, na.rm=TRUE)

## create toy HydroST.raster
s <- stack(lapply(1:12, FUN=function(x)
             raster(matrix(data=sample(1:5, 64, replace=TRUE), nrow=8))))
plot(s)

t <- seq(from=as.POSIXct("2010-01-01 06:00:00"), by="1 hour", length.out=6)
x <- HydroSTF(data=s, time=t, z=c(0,1), endz=c(1,2))
x

## aggregate in space
aggregate(x, fact=2, FUN=mean)
aggregate(x, fact=4, FUN=mean)
aggregate(x, by="space", FUN=sum)

## aggregate in time
aggregate(x, by="2 hours", FUN=mean)
aggregate(x, by="time", FUN=sum)
```

Arith *Arithmetic*

Description

Basic arithmetic for computations with HydroST objects.

Usage

```
Arith(e1, e2)

## S4 method for signature HydroSTxDF,HydroSTxDF
Arith(e1, e2)

## S4 method for signature HydroSTxDF,numeric
Arith(e1, e2)

## S4 method for signature numeric,HydroSTxDF
Arith(e1, e2)

## S4 method for signature HydroSTF.raster,HydroSTF.raster
Arith(e1, e2)
```

```
## S4 method for signature HydroSTF.raster,numeric
Arith(e1, e2)

## S4 method for signature numeric,HydroSTF.raster
Arith(e1, e2)

## S4 method for signature HydroSTF.array,HydroSTF.array
Arith(e1, e2)

## S4 method for signature HydroSTF.array,numeric
Arith(e1, e2)

## S4 method for signature numeric,HydroSTF.array
Arith(e1, e2)

## S4 method for signature HydroSTxDF,Spatial
Arith(e1, e2)

## S4 method for signature Spatial,HydroSTxDF
Arith(e1, e2)

## S4 method for signature Spatial,numeric
Arith(e1, e2)

## S4 method for signature numeric,Spatial
Arith(e1, e2)
```

Arguments

e1,e2	numeric, Spatial, xts or HydroST objects

Value

A HydroST object

See Also

methods::S4groupGeneric

Examples

```
pts <- data.frame(x = 1:5, y = 1:5)
coordinates(pts) <- ~x+y
ts  <- seq(as.POSIXct("2000-01-01", tz="GMT"), by="1 hour", length.out=24)
dat <- data.frame(data=runif(120))
x   <- HydroSTF(STFDF(sp=pts, time=ts, data=dat))

x
x + x
x * x
x ^ x
x %% x
x %/% x
x / x
```

`as.spacetime` *Coercion to spacetime classes*

Description

Coerce HydroST objects to their closest spacetime equivalent.

Usage

```
as.spacetime(x)

## S4 method for signature HydroSTF.array
as.spacetime(x)

## S4 method for signature HydroSTF.raster
as.spacetime(x)

## S4 method for signature HydroSTFDF
as.spacetime(x)

## S4 method for signature HydroSTSDF
as.spacetime(x)

## S4 method for signature HydroSTIDF
as.spacetime(x)

## S4 method for signature HydroSTF
as.spacetime(x)

## S4 method for signature ST
as.spacetime(x)
```

Arguments

`x`	HydroST object

Details

This function is a shortcut to the coercion methods defined for each of the HydroST classes.

`HydroSTF* --> STFDF`

`HydroSTI* --> STIDF`

`HydroSTS* --> STSDF`

Value

A spacetime object

Examples

```
pts <- data.frame(x = 1:5, y = 1:5)
coordinates(pts) <- ~x+y
ts  <- seq(as.POSIXct("2000-01-01"), by="1 hour", length.out=24)
dat <- data.frame(data=runif(120))
x   <- HydroSTF(STFDF(sp=pts, time=ts, data=dat))
as.spacetime(x)

s <- stack(lapply(1:12, FUN=function(x)
            raster(matrix(data=sample(1:5, 64, replace=TRUE), nrow=8))))
t <- seq(from=as.POSIXct("2010-01-01 06:00:00"), by="1 hour", length.out=6)
x <- HydroSTF(data=s, time=t, z=c(0,1), endz=c(1,2))
as.spacetime(x)
```

`compare` *Compare spatial objects*

Description

Evaluate whether two or more spatial objects (Spatial, Raster, spacetime) have the same extent, coordinates, projection, resolution (if Raster*, SpatialGrid or SpatialPixels), data dimensions and values.

Usage

```
compare(x, ...)

## S4 method for signature Spatial
compare(x, ...)

## S4 method for signature SpatialGrid
compare(x, ...)

## S4 method for signature SpatialPixels
compare(x, ...)

## S4 method for signature ST
compare(x, ..., extent = TRUE, coords = TRUE, crs = TRUE,
  res = TRUE, data = TRUE, values = FALSE, time = TRUE, index = TRUE,
  stopiffalse = TRUE, showwarning = FALSE)

## S4 method for signature HydroSTF.raster
compare(x, ..., extent = TRUE, rowcol = TRUE,
  crs = TRUE, res = FALSE, orig = FALSE, rotation = TRUE,
  values = FALSE, stopiffalse = TRUE, showwarning = FALSE)
```

Arguments

`x`	Spatial*, Raster* or ST* object
`extent`	logical. If TRUE, bounding boxes are compared
`coords`	logical. If TRUE, coordinates are compared

crs	logical. If TRUE, coordinate reference systems are compared
res	logical. If TRUE, resolutions are compared. Only relevant for SpatialGrid and SpatialPixels objects
data	logical. If TRUE, data dimensions are compared
values	logical. If TRUE, values are compared
time	logical. If TRUE, time and endTime is compared. Only relevant for ST* objects
index	logical. If TRUE and 'x' is an STSDF or HydroST.STSDF object then the object index is compared. See spacetime::STSDF for more details
stopiffalse	logical. If TRUE, an error will occur if objects are not the same
showwarning	logical. If TRUE, a warning will be given if objects are not the same. Only relevant when stopiffalse is TRUE.
rowcol	logical. If TRUE, number of rows and columns of the objects are compared
orig	logical. If TRUE, origins are compared
rotation	logical. If TRUE, rotations are compared
...	additional Spatial*, Raster* or ST* objects

Details

The structure of this function is based on raster::compareRaster.

Value

Logical

See Also

compareRaster

Examples

```
pts <- data.frame(x = 1:5, y = 1:5)
coordinates(pts) <- ~x+y
ts  <- seq(as.POSIXct("2000-01-01", tz="GMT"), by="1 hour", length.out=24)
dat <- data.frame(data=runif(120))
x   <- HydroSTF(STFDF(sp=pts, time=ts, data=dat))

compare(x, x, values=TRUE)

y <- x * 10
compare(x, y, values=TRUE, stopiffalse=FALSE)

x1 <- x * 10
compare(x, x1)
compare(x, x1, values=TRUE, stopiffalse=FALSE)

s <- stack(lapply(1:12, FUN=function(x)
            raster(matrix(data=sample(1:5, 48, replace=TRUE), nrow=6))))
t <- seq(from=as.POSIXct("2010-01-01 06:00:00"), by="1 hour", length.out=6)
x <- HydroSTF(data=s, time=t, z=c(0,1), endz=c(1,2))

compare(x, x, values=TRUE)
```

```
## comparing different object classes returns FALSE
y <- as(x, "HydroSTF.array")
compare(x, y, stopiffalse=FALSE)
```

`crop` *Crop gridded data*

Description

Crop gridded HydroST objects. A wrapper for `raster::crop`.

Usage

```
crop(x, y, ...)

## S4 method for signature HydroSTF.raster
crop(x, y, ...)

## S4 method for signature HydroSTF.array
crop(x, y, ...)

## S4 method for signature HydroSTFDF
crop(x, y, ...)
```

Arguments

`x`	A HydroST object
`y`	Extent object, or any object from which an Extent object can be extracted (see `raster::crop`)
`...`	additional arguments to `raster::crop`

Value

A HydroST object

See Also

`raster::crop`

Examples

```
s  <- stack(lapply(1:12, FUN=function(x)
             raster(matrix(data=sample(1:5, 64, replace=TRUE), nrow=8))))
t  <- seq(from=as.POSIXct("2010-01-01 06:00:00"), by="1 hour", length.out=6)
x  <- HydroSTF(data=s, time=t, z=c(0,1), endz=c(1,2))

x1 <- crop(x, extent(c(0,0.5,0,0.5)))
extent(x1)

y  <- as(x, "HydroSTF.array")
y1 <- crop(x, extent(c(0,0.5,0,0.5)))

all.equal(x1, as(y1, "HydroSTF.raster"))
```

crs *Coordinate reference system*

Description

Retrieve the coordinate reference system (CRS) of a HydroST object. A wrapper for `raster::crs`.

Usage

```
crs(x, ...)

## S4 method for signature ST
crs(x, asText = FALSE, ...)
```

Arguments

`x`	object inheriting from ST
`asText`	logical. If 'TRUE', the projection is returned as text. Otherwise a 'CRS' object is returned
`...`	additional arguments (none)

Value

CRS or character

See Also

`raster::crs`

Examples

```
data(plyn_aws_data)
prec <- HydroSTF(plyn_aws_data[,,"RAINFALL",drop=FALSE])
crs(prec)
crs(prec, asText=TRUE)
```

diff *Time differences*

Description

Time differences between time and endTime slots in objects inheriting from class ST.

Usage

```
## S3 method for class ST
diff(x, lag, differences, ...)
```

Arguments

x	Object inheriting from class ST (including HydroST objects)
lag	not used. Included for consistency with S3 generic
differences	not used. Included for consistency with S3 generic
...	additional arguments (none)

Value

POSIXct

Examples

```
pts <- data.frame(x = 1:5, y = 1:5)
coordinates(pts) <- ~x+y
ts  <- seq(as.POSIXct("2000-01-01", tz="GMT"), by="1 hour", length.out=24)
dat <- data.frame(data=runif(120))
x   <- HydroSTF(STFDF(sp=pts, time=ts, data=dat))

diff(x)
```

---

dim *Dimensions of a HydroST object*

Description

Get the dimensions of a HydroST object (space, time, depth, variables).

Usage

```
## S3 method for class HydroSTF
dim(x)

## S3 method for class HydroSTS
dim(x)

## S3 method for class HydroSTI
dim(x)
```

Arguments

x	HydroST object

Value

numeric

Examples

```
pts <- data.frame(x = 1:5, y = 1:5)
coordinates(pts) <- ~x+y
ts  <- seq(as.POSIXct("2000-01-01", tz="GMT"), by="1 hour", length.out=24)
dat <- data.frame(data=runif(120))
x   <- HydroSTF(STFDF(sp=pts, time=ts, data=dat))
dim(x)
```

`extent` *Extent*

Description

Return the spatial extent of a HydroST object. A wrapper for `raster::extent`.

Usage

```
extent(x, ...)

## S4 method for signature ST
extent(x, ...)

## S4 method for signature HydroSTF.raster
extent(x, ...)
```

Arguments

`x`	a HydroST object
`...`	additional arguments (none)

Value

Extent object

See Also

`raster::extent`

Examples

```
s  <- stack(lapply(1:12, FUN=function(x)
             raster(matrix(data=sample(1:5, 64, replace=TRUE), nrow=8),
                    xmn=1, xmx=9, ymn=1, ymx=9)))
t  <- seq(from=as.POSIXct("2010-01-01 06:00:00"), by="1 hour", length.out=6)
x  <- HydroSTF(data=s, time=t, z=c(0,1), endz=c(1,2))
extent(x)
```

`extract` *Extract values*

Description

Extract values from a gridded HydroST object at the location of other spatial data.

Usage

```
extract(x, y, ...)

## S4 method for signature HydroSTF.raster,SpatialPoints
extract(x, y, ...)

## S4 method for signature HydroSTF.raster,SpatialLines
extract(x, y, ...)

## S4 method for signature HydroSTF.raster,SpatialPolygons
extract(x, y, ...)

## S4 method for signature HydroSTF.raster,data.frame
extract(x, y, ...)

## S4 method for signature HydroSTF.raster,matrix
extract(x, y, ...)

## S4 method for signature HydroSTF.raster,numeric
extract(x, y, ...)

## S4 method for signature HydroSTF,ANY
extract(x, y, ...)

## S4 method for signature HydroSTF,ST
extract(x, y, ...)
```

Arguments

x	HydroSTF.raster object, or an object that can be coerced to HydroSTF.raster
y	points represented by a two-column matrix or data.frame, a SpatialPoints, SpatialPolygons or SpatialLines object, or a numeric vector representing cell numbers
...	additional arguments to `raster::extract`

Details

The data slot of x is coerced to a RasterStack and passed to `raster::extract` with the sp slot of y.

Value

A HydroST object

Examples

```
s  <- stack(lapply(1:12, FUN=function(x)
             raster(matrix(data=sample(1:5, 64, replace=TRUE), nrow=8),
                    xmn=1, xmx=9, ymn=1, ymx=9)))
t  <- seq(from=as.POSIXct("2010-01-01 06:00:00"), by="1 hour", length.out=6)
x  <- HydroSTF(data=s, time=t, z=c(0,1), endz=c(1,2))

pts <- data.frame(x=sample(1:8, 5), y=sample(1:8, 5))
x1  <- extract(x, pts)

pts <- SpatialPoints(pts)
x2  <- extract(x, pts)

ts  <- seq(as.POSIXct("2000-01-01", tz="GMT"), by="1 hour", length.out=24)
dat <- data.frame(data=runif(120))
y   <- HydroSTF(STFDF(sp=pts, time=ts, data=dat))
x3  <- extract(x, y)

all.equal(x1, x2)
all.equal(x1, x3)
```

`Extract by index` *Extract by index*

Description

Subset objects belonging to classes defined in package Hydro by index.

Usage

```
## S4 replacement method for signature HydroCatchment,ANY,missing,HydroST
x[[i, j]] <- value

## S4 method for signature HydroCatchment,ANY,missing
x[[i, j, ...]]

## S4 method for signature HydroCatchment,ANY,ANY
x[i, j, ..., drop = FALSE]

## S4 replacement method for signature HydroSTFDF,ANY,ANY,numeric
x[i, j, ...] <- value

## S4 method for signature HydroSTF.raster,ANY,ANY
x[i, j, ..., drop = FALSE]

## S4 method for signature HydroSTF.array,ANY,ANY
x[i, j, ..., drop = FALSE]

## S4 method for signature HydroSTFDF,ANY,ANY
x[i, j, ..., drop = FALSE]
```

Arguments

x	HydroST or HydroCatchment
i	numeric. Selection of spatial entities
j	numeric. Selection of temporal entities
value	HydroST object
drop	logical. If TRUE the result is coerced to an object with the lowest number of dimensions. For example if a single spatial entity is selected an xts object is returned. If FALSE the returned object has the same class as x
...	selection of attributes

Value

HydroST or HydroCatchment

---

geometry *Retrieve geometry*

Description

Retrieve the geometry of a HydroST* object.

Usage

```
geometry(obj)

## S4 method for signature HydroSTF
geometry(obj)

## S4 method for signature HydroSTI
geometry(obj)

## S4 method for signature HydroSTS
geometry(obj)

## S4 method for signature HydroCatchment
geometry(obj)
```

Arguments

obj	a HydroST object

Value

a HydroSTF, HydroSTI or HydroSTS object, depending on the class of obj

See Also

sp::geometry

Examples

```
pts <- data.frame(x = 1:5, y = 1:5)
coordinates(pts) <- ~x+y
ts  <- seq(as.POSIXct("2000-01-01", tz="GMT"), by="1 hour", length.out=24)
dat <- data.frame(data=runif(120))
x   <- HydroSTF(STFDF(sp=pts, time=ts, data=dat))
geometry(x)
```

`getDataObject` *Get data from HydroCatchment*

Description

Retrieve HydroST object from HydroCatchment object.

Usage

```
getDataObject(x, variablename, variableunitsid = NA, valuetype = NA,
  datatype = NA, ...)
```

Arguments

`x`	HydroCatchment
`variablename`	character
`variableunitsid`	numeric
`valuetype`	character
`datatype`	character
`...`	additional arguments (none)

Value

HydroST

Examples

```
## See HydroCatchment-class documentation
```

gridded *Test whether object is gridded*

Description

Test whether a HydroST object is gridded. A wrapper for sp::gridded.

Usage

```
gridded(obj)

## S4 method for signature ST
gridded(obj)
```

Arguments

obj	An object inheriting from ST

Details

The original definition of the generic in package sp allows users to specify data as gridded by defining a "gridded<-" method. This is not extended to HydroST objects because there are coercion methods that have the same effect.

Value

Logical

See Also

sp::gridded

Examples

```
s  <- stack(lapply(1:12, FUN=function(x)
             raster(matrix(data=sample(1:5, 64, replace=TRUE), nrow=8),
                    xmn=1, xmx=9, ymn=1, ymx=9)))
t  <- seq(from=as.POSIXct("2010-01-01 06:00:00"), by="1 hour", length.out=6)
x  <- HydroSTF(data=s, time=t, z=c(0,1), endz=c(1,2))
gridded(x) ## TRUE
gridded(as(x, "HydroSTFDF")) ## TRUE

pts <- data.frame(x = 1:5, y = 1:5)
y  <- extract(x, pts)
gridded(y) ## FALSE
```

hc *Template HydroCatchment object*

Description

HydroCatchment object with simple area, network and outlet features and empty parameters and data slots.

Usage

```
hc
```

Format

A HydroCatchment object

Examples

```
data(hc)
```

HydroCatchment-class *Class HydroCatchment*

Description

An S4 class for hydrological catchment features.

Usage

```
HydroCatchment(x, ...)

## S4 method for signature ANY
HydroCatchment(x, area, network, outlet, parameters = list(),
  data = list(), ...)

## S4 method for signature HydroCatchment
HydroCatchment(x, ...)

## S4 method for signature HydroCatchment
update(object, area, network, outlet, parameters,
  data, replace = TRUE, ...)

## S3 method for class HydroCatchment
c(..., recursive = FALSE)
```

Arguments

x,object	HydroCatchment object
area	SpatialPolygons
network	SpatialLines
outlet	SpatialPoints
parameters	named list
data	list of HydroST objects
replace	logical indicating whether data object in HydroCatchment should be replaced
recursive	not used. Included for consistency with S3 generic
...	additional objects

Slots

`area` SpatialPolygons to represent catchment boundary

`network` SpatialLines to represent river network

`outlet` SpatialPoints to represent catchment outlet

`parameters` list

`data` list of HydroST objects

Methods

[`signature(x = "HydroCatchment")`: subset HydroCatchment object

Examples

```
data(plyn_aws_data)
var1 <- HydroSTF(plyn_aws_data[,,"SOLAR_RADIATION",drop=FALSE],
                 metadata=list(VariableName="Global Radiation",
                               VariableUnitsID=33,
                               ValueType="Field Observation",
                               DataType="Average"))

var2 <- HydroSTF(plyn_aws_data[,,"NET_RADIATION",drop=FALSE],
                 metadata=list(VariableName="Radiation, net",
                               VariableUnitsID=33,
                               ValueType="Field Observation",
                               DataType="Average"))

var3 <- HydroSTF(plyn_aws_data[,,"WET_BULB_TEMP",drop=FALSE],
                 metadata=list(VariableName="Temperature, wet-bulb",
                               VariableUnitsID=96,
                               ValueType="Field Observation",
                               DataType="Average"))

var4 <- HydroSTF(plyn_aws_data[,,"DRY_BULB_TEMP",drop=FALSE],
                 metadata=list(VariableName="Temperature",
                               VariableUnitsID=96,
                               ValueType="Field Observation",
                               DataType="Average"))

var5 <- HydroSTF(plyn_aws_data[,,"WIND_SPEED",drop=FALSE],
```

```
                  metadata=list(VariableName="Wind speed",
                                VariableUnitsID=119,
                                ValueType="Field Observation",
                                DataType="Average"))

var6 <- HydroSTF(plyn_aws_data[,,"WIND_DIRECTION",drop=FALSE],
                  metadata=list(VariableName="Wind direction",
                                VariableUnitsID=139,
                                ValueType="Field Observation",
                                DataType="Average"))

var7 <- HydroSTF(plyn_aws_data[,,"RAINFALL",drop=FALSE],
                  metadata=list(VariableName="Precipitation",
                                VariableUnitsID=54,
                                ValueType="Field Observation",
                                DataType="Cumulative"))

data(hc)
hc <- update(hc, data=list(var1, var2, var3, var4, var5, var6, var7))
VariableName(hc)

## Extract methods for HydroCatchment:
hc[1]
hc[1,drop=TRUE]

hc[[1]]
hc[["Wind speed"]]

hc[c("Temperature","Precipitation")]

## More precise control over variable selection:
getDataObject(hc,
              variablename="Precipitation",
              variableunitsid=54,
              valuetype="Field Observation",
              datatype="Cumulative")
```

HydroSTF-class *Class HydroSTF*

Description

An S4 class to represent hydrological data with a full space-time grid. HydroSTF inherits from spacetime::STFDF.

Usage

```
HydroSTF(data, ...)

## S4 method for signature missing
HydroSTF(data, sp, time, endTime, z = as.numeric(NA),
  endz = as.numeric(NA), ...)

## S4 method for signature array
```

```
HydroSTF(data, sp, time, endTime, z = as.numeric(NA),
  endz = as.numeric(NA), ...)

## S4 method for signature Raster
HydroSTF(data, time, endTime, z = as.numeric(NA),
  endz = as.numeric(NA), ...)

## S4 method for signature STFDF
HydroSTF(data, z = as.numeric(NA), endz = as.numeric(NA),
  ...)

## S4 method for signature xts
HydroSTF(data, sp, endTime, z = as.numeric(NA),
  endz = as.numeric(NA), ...)

## S4 method for signature HydroSTF,xts
coerce(from,to)
```

Arguments

data	data.frame, array or RasterStack
sp	Spatial
time	xts
endTime	POSIXct
z	numeric
endz	numeric
...	additional arguments (none)
from	HydroSTF
to	target class

Slots

sp an object deriving from class Spatial

time an object of class xts, or a time vector. See spacetime::ST

endTime vector of class POSIXct holding end points of time intervals

z numeric vector with the upper (lower) limit of each layer represented by the data. The values should be supplied as offsets from elevation in meters, such that a value of 0 indicates the earth's surface

endz numeric vector indicating the lower (upper) limit of each layer represented by the data

Methods

coerce HydroST,xts

Examples

```
library(xts)
pts <- data.frame(x = 1:5, y = 1:5)
coordinates(pts) <- ~x+y
ts <- seq(as.POSIXct("2000-01-01", tz="GMT"), by="1 hour", length.out=168)
```

```
ts <- xts(data.frame(timeIndex=seq(1:length(ts))), ts)
stf <- HydroSTF(sp=pts, time=ts, endTime=index(ts) + 3600)
plot(stf)
```

`HydroSTF.array-class` *Class HydroSTF.array*

Description

An S4 class for gridded hydrological data.

Usage

```
## S4 method for signature HydroSTF.array,array
coerce(from,to)

## S4 method for signature HydroSTF.array,STFDF
coerce(from,to)

## S4 method for signature HydroSTF.array,HydroSTFDF
coerce(from,to)

## S4 method for signature HydroSTF.array,RasterStack
coerce(from,to)

## S4 method for signature HydroSTF.array,HydroSTF.raster
coerce(from,to)
```

Arguments

`from`	HydroSTF.array
`to`	target class

Slots

`sp` an object deriving from class Spatial

`time` an object of class xts, or a time vector. See `spacetime::ST`

`endTime` vector of class POSIXct holding end points of time intervals

`z` numeric vector with the upper (lower) limit of each layer represented by the data. The values should be supplied as offsets from elevation in meters, such that a value of 0 indicates the earth's surface

`endz` numeric vector indicating the lower (upper) limit of each layer represented by the data

`data` array with four dimensions in the following order: time, x, y, z

`metadata` list containing object metadata. See `metadata`

Examples

```
arr <- array(data=sample(1:10, 250, replace=TRUE), dim=c(10,5,5,1))
pts <- SpatialPoints(expand.grid(x=seq(279500,283500,by=1000),
                                 y=seq(283500,287500,by=1000)),
                     proj4string=crs(plyn_aws_data))
t <- seq(as.POSIXct("2010-01-01", tz="UTC"), by="1 hour", length.out=10)
x <- HydroSTF(data=arr, sp=pts, time=t)
plot(as(x, "RasterStack"))
```

HydroSTF.raster-class *Class HydroSTF.raster*

Description

An S4 class for gridded hydrological data.

Usage

```
## S4 method for signature HydroSTF.raster,RasterStack
coerce(from,to)

## S4 method for signature HydroSTF.raster,STFDF
coerce(from,to)

## S4 method for signature HydroSTF.raster,HydroSTFDF
coerce(from,to)

## S4 method for signature HydroSTF.raster,HydroSTF.array
coerce(from,to)
```

Arguments

from	HydroSTF.raster
to	target class

Slots

sp an object deriving from class Spatial

time an object of class xts, or a time vector. See spacetime::ST

endTime vector of class POSIXct holding end points of time intervals

z numeric vector with the upper (lower) limit of each layer represented by the data. The values should be supplied as offsets from elevation in meters, such that a value of 0 indicates the earth's surface

endz numeric vector indicating the lower (upper) limit of each layer represented by the data

data RasterStack with z index moving faster than time index

metadata list containing object metadata. See metadata

Examples

```
s <- stack(lapply(1:10, FUN=function(x)
             raster(xmn=279000, xmx=284000, ymn=283000, ymx=288000,
                    res=1000, vals=sample(1:5, 25, replace=TRUE),
                    crs=crs(plyn_aws_data))))
t <- seq(as.POSIXct("2010-01-01", tz="UTC"), by="1 hour", length.out=5)
x <- HydroSTF(data=s, time=t, z=c(0,1), endz=c(1,2))
plot(as(x, "RasterStack"))
```

`HydroSTFDF-class` *Class HydroSTFDF*

Description

An S4 class for hydrological data with full space-time grid. Based on `spacetime::STFDF`.

Usage

```
## S4 method for signature HydroSTFDF,STFDF
coerce(from,to)

## S4 method for signature HydroSTFDF,RasterStack
coerce(from,to)

## S4 method for signature HydroSTFDF,HydroSTF.raster
coerce(from,to)

## S4 method for signature HydroSTFDF,HydroSTF.array
coerce(from,to)
```

Arguments

`from`	HydroSTFDF
`to`	target class

Slots

`sp` an object deriving from class Spatial

`time` an object of class xts, or a time vector. See `spacetime::ST`

`endTime` vector of class POSIXct holding end points of time intervals

`z` numeric vector with the upper (lower) limit of each layer represented by the data. The values should be supplied as offsets from elevation in meters, such that a value of 0 indicates the earth's surface

`endz` numeric vector indicating the lower (upper) limit of each layer represented by the data

`data` data.frame with rows corresponding to observations: see `spacetime::STFDF` for details

`metadata` list containing object metadata. See `metadata`

Examples

```
pts <- SpatialPoints(data.frame(x = 1:5, y = 1:5))
ts  <- seq(as.POSIXct("2000-01-01", tz="GMT"), by="1 hour", length.out=24)
dat <- data.frame(data=runif(120))

x   <- HydroSTF(STFDF(sp=pts, time=ts, data=dat))  ## supply STFDF

library(xts)
dat <- as.data.frame(matrix(data=runif(120), nrow=length(ts))) ## space-wide table
x   <- HydroSTF(xts(dat, ts), sp=pts) ## supply xts
```

HydroSTI-class *Class HydroSTI*

Description

An S4 class to represent hydrological data with an unstructured space-time grid. HydroSTI inherits from `spacetime::STIDF`.

Usage

```
HydroSTI(data, ...)

## S4 method for signature missing
HydroSTI(data, sp, time, endTime, z = as.numeric(NA),
  endz = as.numeric(NA), ...)

## S4 method for signature STIDF
HydroSTI(data, z = as.numeric(NA), endz = as.numeric(NA),
  ...)

## S4 method for signature HydroSTI,xts
coerce(from,to)
```

Arguments

`data`	data.frame
`sp`	Spatial
`time`	xts
`endTime`	POSIXct
`z`	numeric
`endz`	numeric
`...`	additional arguments (none)
`from`	HydroSTI
`to`	target class

Slots

sp an object deriving from class Spatial

time an object of class xts, or a time vector. See spacetime::ST

endTime vector of class POSIXct holding end points of time intervals

z numeric vector with the upper (lower) limit of each layer represented by the data. The values should be supplied as offsets from elevation in meters, such that a value of 0 indicates the earth's surface

endz numeric vector indicating the lower (upper) limit of each layer represented by the data

Examples

```
library(xts)
sp <- SpatialPoints(data.frame(x=sample(10, 10, replace=TRUE),
                               y=sample(10, 10, replace=TRUE)))
ts <- seq(as.POSIXct("2000-01-01", tz="GMT"), by="1 hour", length.out=24)
ts <- ts[sample(10, 10, replace=TRUE)]
ts <- xts(data.frame(timeIndex=seq(1:length(ts))), ts)

sti <- HydroSTI(sp=sp, time=ts, endTime=index(ts) + 3600)
plot(sti)
```

HydroSTIDF-class *Class HydroSTIDF*

Description

An S4 class for unstructured hydrological data. Based on spacetime::STIDF.

Usage

```
## S4 method for signature HydroSTIDF,STIDF
coerce(from,to)
```

Arguments

from	HydroSTIDF
to	target class

Slots

sp an object deriving from class Spatial

time an object of class xts, or a time vector. See spacetime::ST

endTime vector of class POSIXct holding end points of time intervals

z numeric vector with the upper (lower) limit of each layer represented by the data. The values should be supplied as offsets from elevation in meters, such that a value of 0 indicates the earth's surface

endz numeric vector indicating the lower (upper) limit of each layer represented by the data

data data.frame with rows corresponding to observations: see spacetime::STIDF for details

metadata list containing object metadata. See metadata

Examples

```
library(xts)
sp <- SpatialPoints(data.frame(x=sample(10, 10, replace=TRUE),
                               y=sample(10, 10, replace=TRUE)))
ts <- seq(as.POSIXct("2000-01-01", tz="GMT"), by="1 hour", length.out=24)
ts <- ts[sample(10, 10, replace=TRUE)]
dat <- data.frame(data=runif(10))

stidf <- STIDF(sp=sp, time=ts, data=dat)

HydroSTI(stidf)
```

HydroSTS-class *Class HydroSTS*

Description

An S4 class to represent hydrological data with a partial space-time grid. HydroSTS inherits from `spacetime::STSDF`.

Usage

```
HydroSTS(data, ...)

## S4 method for signature missing
HydroSTS(data, sp, time, endTime, index,
  z = as.numeric(NA), endz = as.numeric(NA), ...)

## S4 method for signature STSDF
HydroSTS(data, z = as.numeric(NA), endz = as.numeric(NA),
  ...)

## S4 method for signature HydroSTS,xts
coerce(from,to)
```

Arguments

`data`	data.frame
`sp`	Spatial
`time`	xts
`endTime`	POSIXct
`index`	two-column matrix. See `spacetime::STSDF`.
`z`	numeric
`endz`	numeric
`...`	additional arguments (none)
`from`	HydroSTS
`to`	target class

Slots

sp an object deriving from class Spatial

time an object of class xts, or a time vector. See spacetime::ST

index two-column matrix. See spacetime::STSDF.

endTime vector of class POSIXct holding end points of time intervals

z numeric vector with the upper [lower] limit of each layer represented by the data. The values should be supplied as offsets from elevation in meters, such that a value of 0 indicates the earth's surface

endz numeric vector indicating the lower [upper] limit of each layer represented by the data

Examples

```
library(xts)
ts <- seq(as.POSIXct("2000-01-01", tz="GMT"), by="1 hour", length.out=24)
ts <- xts(data.frame(timeIndex=seq(1:length(ts))), ts)
sp <- SpatialPoints(data.frame(x=sample(1:5, 3, replace=TRUE),
                               y=sample(1:5, 3, replace=TRUE)))

space.index <- rep(1:3, 5)
time.index  <- c(sample(1:5,3),
                 sample(6:10,3),
                 sample(11:15,3),
                 sample(16:20,3),
                 sample(21:24,3))
idx <- matrix(data=c(space.index, time.index), ncol=2)
sts <- HydroSTS(sp=sp, time=ts, endTime=index(ts) + 3600, index=idx)
plot(sts)
```

HydroSTSDF-class *Class HydroSTSDF*

Description

An S4 class for spatio-temporal hydrological data with partial space-time grids. Based on spacetime::STSDF.

Usage

```
## S4 method for signature HydroSTSDF,STSDF
coerce(from,to)
```

Arguments

from	HydroSTSDF
to	target class

Slots

sp an object deriving from class Spatial

time an object of class xts, or a time vector. See spacetime::ST

endTime vector of class POSIXct holding end points of time intervals

z numeric vector with the upper [lower] limit of each layer represented by the data. The values should be supplied as offsets from elevation in meters, such that a value of 0 indicates the earth's surface

endz numeric vector indicating the lower [upper] limit of each layer represented by the data

data data.frame with rows corresponding to observations: see spacetime::STSDF for details

index two-column matrix: see spacetime::STSDF for details

metadata list containing object metadata. See metadata

Examples

```
library(xts)
ts <- seq(as.POSIXct("2000-01-01", tz="GMT"), by="1 hour", length.out=24)
ts <- xts(data.frame(timeIndex=seq(1:length(ts))), ts)
sp <- SpatialPoints(data.frame(x=sample(1:5, 3, replace=TRUE),
                               y=sample(1:5, 3, replace=TRUE)))

space.index <- rep(1:3, 5)
time.index  <- c(sample(1:5,3),
                 sample(6:10,3),
                 sample(11:15,3),
                 sample(16:20,3),
                 sample(21:24,3))
idx <- matrix(data=c(space.index, time.index), ncol=2)
dat <- data.frame(data=runif(15))

stsdf <- STSDF(sp=sp, time=ts, data=dat, index=idx)

HydroSTS(stsdf)
```

---

interval *Interval*

Description

Time interval of regular time series.

Usage

```
interval(x, ...)

## S4 method for signature ST
interval(x, units = "hours", ...)

## S4 method for signature HydroCatchment
interval(x, units = "hours", ...)
```

Arguments

x	object inheriting from ST or HydroCatchment
units	character. See base::difftime
...	additional arguments (none)

Value

difftime

Examples

```
pts <- data.frame(x = 1:5, y = 1:5)
coordinates(pts) <- ~x+y
ts  <- seq(as.POSIXct("2000-01-01", tz="GMT"), by="1 hour", length.out=24)
dat <- data.frame(data=runif(120))
x   <- HydroSTF(STFDF(sp=pts, time=ts, data=dat))

interval(x)
```

Math *Mathematical functions*

Description

Generic mathematical functions that can be used with HydroST objects as an argument.

Usage

```
Math(x)

## S4 method for signature ST
Math(x)

## S4 method for signature HydroSTF.raster
Math(x)
```

Arguments

x	object inheriting from ST

Value

ST object

Examples

```
pts <- data.frame(x = 1:5, y = 1:5)
coordinates(pts) <- ~x+y
ts  <- seq(as.POSIXct("2000-01-01", tz="GMT"), by="1 hour", length.out=24)
dat <- data.frame(data=runif(120))
x   <- HydroSTF(STFDF(sp=pts, time=ts, data=dat))

abs(x)
sign(x)
sqrt(x)
floor(x)
ceiling(x)
log(x)
exp(x)
```

`metadata` *Get or set metadata*

Description

Get or set metadata of a HydroST object.

Usage

```
metadata(x)

## S4 method for signature HydroSTF
metadata(x)

## S4 method for signature HydroCatchment
metadata(x)

## S4 replacement method for signature HydroST
metadata(x) <- value

updateMetadata(x, ...)

## S4 method for signature HydroST
updateMetadata(x, value)

VariableName(x)

## S4 method for signature HydroST
VariableName(x)

## S4 method for signature HydroCatchment
VariableName(x)

VariableName(x) <- value

## S4 replacement method for signature HydroST,character
VariableName(x) <- value
```

```
VariableUnitsID(x)

## S4 method for signature HydroST
VariableUnitsID(x)

## S4 method for signature HydroCatchment
VariableUnitsID(x)

VariableUnitsID(x) <- value

## S4 replacement method for signature HydroST,numeric
VariableUnitsID(x) <- value

ValueType(x)

## S4 method for signature HydroST
ValueType(x)

## S4 method for signature HydroCatchment
ValueType(x)

ValueType(x) <- value

## S4 replacement method for signature HydroST,character
ValueType(x) <- value

DataType(x)

## S4 method for signature HydroST
DataType(x)

## S4 method for signature HydroCatchment
DataType(x)

DataType(x) <- value

## S4 replacement method for signature HydroST,character
DataType(x) <- value
```

Arguments

x	A HydroST object
value	named list containing metadata entries
...	additional arguments (none)

Details

Classes inheriting from HydroST include a slot `metadata`. This is a list containing at least the following elements:

VariableName The variable name, e.g. "Precipitation", "Discharge"

VariableUnitsID The units of the variable, e.g. "millimeters", "cubic meters per second"

ValueType The value type, e.g. "Field Observation", "Model Simulation Result"

DataType The (statistical) data type, e.g. "Cumulative", "Average"

These entries must correspond with the CUAHSI controlled vocabulary (http://his.cuahsi.org/mastercvdata.html).

Value

list or HydroST

Examples

```
data(plyn_aws_data)
x <- HydroSTF(plyn_aws_data[,,"DRY_BULB_TEMP",drop=FALSE])

metadata(x)
metadata <- list(VariableName="Temperature",
                 VariableUnitsID=96,
                 ValueType="Field Observation",
                 DataType="Unknown")
metadata(x) <- metadata
metadata(x)

y <- HydroSTF(plyn_aws_data[,,"WET_BULB_TEMP",drop=FALSE])
metadata(y) <- modifyList(metadata(x), list(VariableName="Temperature, wet-bulb"))
metadata(y)

VariableName(x)
VariableUnitsID(x)
ValueType(x)
DataType(x)
```

---

names *Names of HydroST objects*

Description

Get names of HydroST objects.

Usage

```
## S4 method for signature HydroST
names(x)

## S4 method for signature HydroCatchment
names(x)
```

Arguments

x	HydroST object or HydroCatchment

Value

Character

over *Spatio-temporal overlay*

Description

Consistent spatio-temporal overlay for objects inheriting from HydroST. These methods extend those defined in package spacetime and sp.

Usage

```
over(x, y, returnList = FALSE, fn = NULL, ...)

## S4 method for signature ST,xts
over(x, y, returnList = FALSE, fn = NULL, ...)
```

Arguments

x	geometry (S/T locations) of the queries
y	layer from which the geometries or attributes are queried
returnList	logical; determines whether a list is returned, or an index vector
fn	(optional) a function; see value
...	arguments passed on to function fn

Value

If x inherits from HydroST.ST, a vector, data.frame or list as described in over::over

If x is a HydroST.raster or HydroST.array object a list with elements 'space' and 'time'. The 'space' vector has length equal to ncell(x@data) and the 'time' vector, which is expanded according to the depth dimension, has length equal to nlayers(x@data).

See Also

over::over, over::over

Examples

```
data(plyn_aws_data)
x1 <- HydroSTF(plyn_aws_data[,,"DRY_BULB_TEMP",drop=FALSE])
x2 <- x1[1:2,]

## HydroST,HydroST-method
over(x1, x2)

s  <- stack(lapply(1:12, FUN=function(x)
             raster(matrix(data=sample(1:5, 64, replace=TRUE), nrow=8),
                    xmn=1, xmx=9, ymn=1, ymx=9)))
t  <- seq(from=as.POSIXct("2010-01-01 06:00:00"), by="1 hour", length.out=6)
x1 <- HydroSTF(data=s, time=t, z=c(0,1), endz=c(1,2))

## HydroST.raster,HydroST-method
x2 <- x1[,1:3]
```

```
over(x1, x2)

## HydroSTF.array,HydroSTF-method
x1 <- as(x1, "HydroSTF.array")

ix <- over(x1, x2)
```

---

plyn_aws_data *Plynlimon meteorological data*

Description

Space-time dataset of meteorological variables.

Usage

```
plyn_aws_data
```

Format

An STFDF object containing the following variables:

[,1] SITE

[,2] HOUR_ENDED

[,3] SOLAR_RADIATION

[,4] NET_RADIATION

[,5] WET_BULB_TEMP

[,6] DRY_BULB_TEMP

[,7] WIND_SPEED

[,8] WIND_DIRECTION

[,9] RAINFALL

Source

Centre for Ecology and Hydrology

Examples

```
data(plyn_aws_data)
```

plyn_flow_data *Plynlimon flow data*

Description

Space-time dataset of stream discharge.

Usage

```
plyn_flow_data
```

Format

An STFDF object containing the following variables:

[,1] DISCHARGE

Source

Centre for Ecology and Hydrology

Examples

```
data(plyn_flow_data)
```

regular *Test for regular time steps*

Description

Test whether a HydroST object has regular or irregular time steps.

Usage

```
regular(x, ...)

irregular(x, ...)
```

Arguments

x	HydroST object
...	additional arguments (none)

Value

logical

Examples

```
pts <- data.frame(x = 1:5, y = 1:5)
coordinates(pts) <- ~x+y
ts  <- seq(as.POSIXct("2000-01-01", tz="GMT"), by="1 hour", length.out=24)
dat <- data.frame(data=runif(120))
x   <- HydroSTF(STFDF(sp=pts, time=ts, data=dat))

regular(x)
```

`resample` *Resample HydroST objects*

Description

Resample HydroST objects with gridded data. A wrapper for `raster::resample`.

Usage

```
resample(x, y, ...)

## S4 method for signature HydroSTF.raster,Raster
resample(x, y, ...)
```

Arguments

`x`	HydroST object to be resampled
`y`	object with parameters to which 'x' should be resampled. This can be a HydroST object that can be coerced to HydroST.raster, a SpatialGrid or SpatialPixels object or a Raster* object
`...`	additional arguments to `raster::resample`

Value

HydroST object

See Also

`raster::resample`

Examples

```
s <- stack(lapply(1:12, FUN=function(x)
            raster(matrix(data=sample(1:5, 64, replace=TRUE), nrow=8),
                   xmn=1, xmx=9, ymn=1, ymx=9)))
t <- seq(from=as.POSIXct("2010-01-01 06:00:00"), by="1 hour", length.out=6)
x <- HydroSTF(data=s, time=t, z=c(0,1), endz=c(1,2))

r <- raster(xmn=1, xmx=9, ymn=1, ymx=9, res=0.2)
y <- resample(x, r)

dim(x)
dim(y)
```

spTransform *Map projection and datum transformation*

Description

Map projection and datum transformation. A wrapper for sp::spTransform.

Usage

```
spTransform(x, CRSobj, ...)

## S4 method for signature HydroSTF,ANY
spTransform(x, CRSobj, ...)

## S4 method for signature HydroSTI,ANY
spTransform(x, CRSobj, ...)

## S4 method for signature HydroSTS,ANY
spTransform(x, CRSobj, ...)

## S4 method for signature HydroSTF.raster,ANY
spTransform(x, CRSobj, ...)

## S4 method for signature HydroSTF.array,ANY
spTransform(x, CRSobj, ...)
```

Arguments

x	HydroST object to be transformed
CRSobj	object of class CRS or character, in which case it is converted to CRS
...	additional arguments (none)

Value

HydroST

See Also

sp::spTransform

Examples

```
s <- stack(lapply(1:12, FUN=function(x)
            raster(xmn=279000, xmx=284000, ymn=283000, ymx=288000,
                   res=1000, vals=sample(1:5, 25, replace=TRUE),
                   crs=crs(plyn_aws_data))))
t <- seq(from=as.POSIXct("2010-01-01 06:00:00"), by="1 hour", length.out=6)
x <- HydroSTF(data=s, time=t, z=c(0,1), endz=c(1,2))
y <- spTransform(x, CRS("+proj=longlat +datum=WGS84")) ## method dispatch overlap?

isLonLat(crs(x))
isLonLat(crs(y))
```

units *Units*

Description

Get units of a HydroST object.

Usage

```
## S3 method for class HydroST
units(x, ...)
```

Arguments

x	HydroST object
...	additional arguments (none)

Value

character

Examples

```
data(plyn_aws_data)
x <- HydroSTF(plyn_aws_data[,,"SOLAR_RADIATION",drop=FALSE],
              metadata=list(VariableName="Global Radiation",
                            VariableUnitsID=33,
                            ValueType="Field Observation",
                            DataType="Average"))
units(x)
```

window *Time windows*

Description

Extract time windows of HydroST objects.

Usage

```
## S3 method for class HydroST
window(x, start = NULL, end = NULL, frequency = NULL,
  deltat = NULL, extend = NULL, ...)
```

Arguments

x	A HydroST object
start	start time of the period of interest in POSIXt format
end	end time of the period of interest in POSIXt format
frequency	not used. Included for consistency with S3 generic
deltat	not used. Included for consistency with S3 generic
extend	not used. Included for consistency with S3 generic
...	additional arguments (none)

Value

A HydroST object

See Also

window.zoo

Examples

```
data(plyn_aws_data)
x <- HydroSTF(plyn_aws_data[,,"DRY_BULB_TEMP",drop=FALSE])
start(x)
end(x)

tz <- attr(index(x@time), "tzone")
y <- window(x,
            start=as.POSIXct("2008-01-01 00:00:00", tz=tz),
            end=as.POSIXct("2008-01-01 23:00:00", tz=tz))
start(y)
end(y)
```

Index

∗Topic **datasets**
hc, 19
plyn_aws_data, 36
plyn_flow_data, 37
[,HydroCatchment,ANY,ANY-method (Extract by index), 15
[,HydroSTF.array,ANY,ANY-method (Extract by index), 15
[,HydroSTF.raster,ANY,ANY-method (Extract by index), 15
[,HydroSTFDF,ANY,ANY-method (Extract by index), 15
[<-,HydroSTFDF,ANY,ANY,numeric-method (Extract by index), 15
[[,HydroCatchment,ANY,missing-method (Extract by index), 15
[[<-,HydroCatchment,ANY,missing,HydroST-method (Extract by index), 15

aggregate, 3, *4*
Arith, 5
Arith,HydroSTF.array,HydroSTF.array-method (Arith), 5
Arith,HydroSTF.array,numeric-method (Arith), 5
Arith,HydroSTF.raster,HydroSTF.raster-method (Arith), 5
Arith,HydroSTF.raster,numeric-method (Arith), 5
Arith,HydroSTxDF,HydroSTxDF-method (Arith), 5
Arith,HydroSTxDF,numeric-method (Arith), 5
Arith,HydroSTxDF,Spatial-method (Arith), 5
Arith,numeric,HydroSTF.array-method (Arith), 5
Arith,numeric,HydroSTF.raster-method (Arith), 5
Arith,numeric,HydroSTxDF-method (Arith), 5
Arith,numeric,Spatial-method (Arith), 5
Arith,Spatial,HydroSTxDF-method (Arith), 5
Arith,Spatial,numeric-method (Arith), 5
as.spacetime, 7
as.spacetime,HydroSTF-method (as.spacetime), 7
as.spacetime,HydroSTF.array-method (as.spacetime), 7
as.spacetime,HydroSTF.raster-method (as.spacetime), 7
as.spacetime,HydroSTFDF-method (as.spacetime), 7
as.spacetime,HydroSTIDF-method (as.spacetime), 7
as.spacetime,HydroSTSDF-method (as.spacetime), 7
as.spacetime,ST-method (as.spacetime), 7

c.HydroCatchment (HydroCatchment-class), 19
coerce,HydroST.STSDF,STSDF-method (HydroSTSDF-class), 29
coerce,HydroSTF,xts-method (HydroSTF-class), 21
coerce,HydroSTF.array,array-method (HydroSTF.array-class), 23
coerce,HydroSTF.array,HydroSTF.raster-method (HydroSTF.array-class), 23
coerce,HydroSTF.array,HydroSTFDF-method (HydroSTF.array-class), 23
coerce,HydroSTF.array,RasterStack-method (HydroSTF.array-class), 23
coerce,HydroSTF.array,STFDF-method (HydroSTF.array-class), 23
coerce,HydroSTF.raster,HydroSTF.array-method (HydroSTF.raster-class), 24
coerce,HydroSTF.raster,HydroSTFDF-method (HydroSTF.raster-class), 24
coerce,HydroSTF.raster,RasterStack-method (HydroSTF.raster-class), 24
coerce,HydroSTF.raster,STFDF-method (HydroSTF.raster-class), 24
coerce,HydroSTFDF,HydroSTF.array-method (HydroSTFDF-class), 25
coerce,HydroSTFDF,HydroSTF.raster-method (HydroSTFDF-class), 25

coerce,HydroSTFDF,RasterStack-method (HydroSTFDF-class), 25
coerce,HydroSTFDF,STFDF-method (HydroSTFDF-class), 25
coerce,HydroSTI,xts-method (HydroSTI-class), 26
coerce,HydroSTIDF,STIDF-method (HydroSTIDF-class), 27
coerce,HydroSTS,xts-method (HydroSTS-class), 28
coerce,HydroSTSDF,STSDF-method (HydroSTSDF-class), 29
compare, 8
compare,HydroSTF.raster-method (compare), 8
compare,Spatial-method (compare), 8
compare,SpatialGrid-method (compare), 8
compare,SpatialPixels-method (compare), 8
compare,ST-method (compare), 8
compareRaster, *9*
crop, *10*, 10
crop,HydroSTF.array,ANY-method (crop), 10
crop,HydroSTF.array-method (crop), 10
crop,HydroSTF.raster,ANY-method (crop), 10
crop,HydroSTF.raster-method (crop), 10
crop,HydroSTFDF,ANY-method (crop), 10
crop,HydroSTFDF-method (crop), 10
crs, *11*, 11
crs,ST-method (crs), 11

DataType (metadata), 32
DataType,HydroCatchment-method (metadata), 32
DataType,HydroST-method (metadata), 32
DataType<- (metadata), 32
DataType<-,HydroST,character-method (metadata), 32
DataType<-,HydroST-method (metadata), 32
diff, 11
difftime, *31*
dim, 12

extent, *13*, 13
extent,HydroSTF.raster-method (extent), 13
extent,ST-method (extent), 13
extract, *14*, 14
Extract by index, 15
extract,HydroSTF,ANY-method (extract), 14
extract,HydroSTF,ST-method (extract), 14
extract,HydroSTF.raster,data.frame-method (extract), 14
extract,HydroSTF.raster,matrix-method (extract), 14
extract,HydroSTF.raster,numeric-method (extract), 14
extract,HydroSTF.raster,SpatialLines-method (extract), 14
extract,HydroSTF.raster,SpatialPoints-method (extract), 14
extract,HydroSTF.raster,SpatialPolygon-method (extract), 14
extract,HydroSTF.raster,SpatialPolygons-method (extract), 14

geometry, *16*, 16
geometry,HydroCatchment-method (geometry), 16
geometry,HydroSTF-method (geometry), 16
geometry,HydroSTI-method (geometry), 16
geometry,HydroSTS-method (geometry), 16
getDataObject, 17
gridded, *18*, 18
gridded,ST-method (gridded), 18

hc, 19
Hydro-package, 3
HydroCatchment (HydroCatchment-class), 19
HydroCatchment,ANY-method (HydroCatchment-class), 19
HydroCatchment,HydroCatchment-method (HydroCatchment-class), 19
HydroCatchment-class, 19
HydroSTF (HydroSTF-class), 21
HydroSTF,array-method (HydroSTF-class), 21
HydroSTF,missing-method (HydroSTF-class), 21
HydroSTF,Raster-method (HydroSTF-class), 21
HydroSTF,STFDF-method (HydroSTF-class), 21
HydroSTF,xts-method (HydroSTF-class), 21
HydroSTF-class, 21
HydroSTF.array-class, 23
HydroSTF.raster-class, 24
HydroSTFDF-class, 25
HydroSTI (HydroSTI-class), 26
HydroSTI,missing-method (HydroSTI-class), 26

HydroSTI,STIDF-method (HydroSTI-class), 26
HydroSTI-class, 26
HydroSTIDF-class, 27
HydroSTS (HydroSTS-class), 28
HydroSTS,missing-method (HydroSTS-class), 28
HydroSTS,STSDF-method (HydroSTS-class), 28
HydroSTS-class, 28
HydroSTSDF-class, 29

interval, 30
interval,HydroCatchment-method (interval), 30
interval,ST-method (interval), 30
irregular (regular), 37

Math, 31
Math,HydroSTF.raster-method (Math), 31
Math,ST-method (Math), 31
metadata, *23–25*, *27*, *30*, 32
metadata,HydroCatchment-method (metadata), 32
metadata,HydroSTF-method (metadata), 32
metadata<-,HydroST-method (metadata), 32

names, 34
names,HydroCatchment-method (names), 34
names,HydroST-method (names), 34

over, *4*, *35*, 35
over,ST,xts-method (over), 35

plyn_aws_data, 36
plyn_flow_data, 37

regular, 37
resample, *38*, 38
resample,HydroSTF.raster,Raster-method (resample), 38

S4groupGeneric, *6*
spTransform, *39*, 39
spTransform,HydroSTF,ANY-method (spTransform), 39
spTransform,HydroSTF-method (spTransform), 39
spTransform,HydroSTF.array,ANY-method (spTransform), 39
spTransform,HydroSTF.array-method (spTransform), 39
spTransform,HydroSTF.raster,ANY-method (spTransform), 39
spTransform,HydroSTF.raster-method (spTransform), 39
spTransform,HydroSTI,ANY-method (spTransform), 39
spTransform,HydroSTI-method (spTransform), 39
spTransform,HydroSTS,ANY-method (spTransform), 39
spTransform,HydroSTS-method (spTransform), 39
ST, *22–25*, *27*, *29*, *30*
STFDF, *21*, *25*
STIDF, *26*, *27*
STSDF, *9*, *28–30*

units, 40
update,ANY-method (HydroCatchment-class), 19
update,HydroCatchment-method (HydroCatchment-class), 19
updateMetadata (metadata), 32
updateMetadata,HydroST-method (metadata), 32

ValueType (metadata), 32
ValueType,HydroCatchment-method (metadata), 32
ValueType,HydroST-method (metadata), 32
ValueType<- (metadata), 32
ValueType<-,HydroST,character-method (metadata), 32
VariableName (metadata), 32
VariableName,HydroCatchment-method (metadata), 32
VariableName,HydroST-method (metadata), 32
VariableName<- (metadata), 32
VariableName<-,HydroST,character-method (metadata), 32
VariableUnitsID (metadata), 32
VariableUnitsID,HydroCatchment-method (metadata), 32
VariableUnitsID,HydroST-method (metadata), 32
VariableUnitsID<- (metadata), 32
VariableUnitsID<-,HydroST,numeric-method (metadata), 32

window, 40