



UNIVERSITETET I BERGEN



Inferring Vegetation Memory from Remote Sensing Data using novel Climate Reconstruction Products

Master's Thesis (M.Sc.)

31/05/2019

at Universitetet i Bergen

In Partial Fulfilment of the Requirements for the Degree Master of Science (M.Sc.)

Primary Supervisor: Dr. Alistair Seddon
Secondary Supervisor: Dr. Richard Davy

by
Erik Kusch
born 03/01/1995 in Eilenburg, Germany
Enrolment number: 239977

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Acknowledgements

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Contents

1	Introduction	1
1.1	Vegetation Memory	2
1.1.1	Terminology	2
1.1.2	Ecological Relevance	3
1.1.3	Quantifying Vegetation Memory	3
1.2	Plant Functional Traits	4
1.2.1	Terminology	5
1.2.2	Ecological Relevance	6
1.2.3	Unifying Information on Plant Functional Traits	6
1.2.4	Putting Plant Function on the Map	7
1.3	Thesis Outline	8
2	Material & Methods	9
2.1	Data	9
2.1.1	Vegetation Indices	9
2.1.1.1	Background	9
2.1.1.2	Normalised Difference Vegetation Index (NDVI)	9
2.1.2	Abiotic Data	11
2.1.2.1	European Centre for Medium-range Weather Forecasts ReAnalysis 5 (ERA5)	11
2.1.2.2	Digital Elevation Models	14
2.1.3	Plant Functional Trait Data	15
2.1.3.1	TRY	15
2.1.3.2	COMPADRE	16
2.1.4	Floral Data	18
2.2	Study Regions	19
2.2.1	Iberian Region	19
2.2.2	Caatinga, Brazil	20
2.2.3	Australia	21
2.3	Analyses	22
2.3.1	Identifying Vegetation Memory	22
2.3.1.1	Data Preparation	22
2.3.1.2	Memory Models	24
2.3.2	Linking Vegetation Memory and Plant Function	30
2.3.2.1	Plant Functional Traits	30
2.3.2.2	Life History Traits	30

3 Results	31
3.1 Iberian Region	31
3.1.1 Vegetation Memory	31
3.1.1.1 Memory Models	32
3.1.1.2 Memory Coefficients	40
3.1.1.3 Vegetation Memory Sensitivity	40
3.1.2 Plant Function	41
3.1.2.1 Plant Functional Traits	41
3.1.2.2 Life History Traits	42
3.2 Caatinga, Brazil	43
3.2.1 Vegetation Memory	43
3.2.2 Plant Function	43
3.3 Australia	43
3.3.1 Vegetation Memory	43
3.3.2 Plant Function	43
3.4 Global Effects	43
3.4.1 Vegetation Memory	43
3.4.2 Plant Function	43
4 Discussion	44
4.1 Iberian Region	44
4.1.1 Vegetation Memory	44
4.1.2 Plant Function	44
4.2 Caatinga, Brazil	44
4.2.1 Vegetation Memory	44
4.2.2 Plant Function	44
4.3 Australia	44
4.3.1 Vegetation Memory	44
4.3.2 Plant Function	44
4.4 Global Effects	44
4.4.1 Vegetation Memory	44
4.4.2 Plant Function	44
5 Conclusion	45
Bibliography	45
A Appendix	54
A.1 Project Requirements	54
A.1.1 R Requirements	54
A.1.1.1 R Packages	54
A.1.1.2 Project Directories	55
A.1.1.3 R Functions for Region Selection and Raster Names	56
A.1.2 ERA5 Data	57
A.1.2.1 Obtaining ERA5 Data	57
A.1.2.2 Pre-Processing ERA5 Data	57
A.2 Data	61
A.2.1 HWSD Data	61

A.2.2	Study Regions	63
A.2.2.1	Iberian Region	63
A.2.2.2	Caatinga, Brazil	65
A.2.2.3	Australia	67
A.3	R Analyses Codes	69
A.3.1	Master File	69
A.3.2	Vegetation Memory	74
A.3.2.1	GIMMs Data	74
A.3.2.2	ERA5	75
A.3.2.3	Calculation of Vegetation Memory	78
A.3.3	Plant Functional Traits	84
A.3.4	COMPADRE	89
A.4	R Data Visualisation Codes	90
A.4.1	Preamble	90
A.4.2	Data Overview	90
A.4.2.1	NDVI	90
A.4.2.2	ERA5	90
A.4.2.3	HWSD	91
A.4.2.4	TRY	92
A.4.2.5	COMPADRE	92
A.4.2.6	Study Regions	93
A.4.3	Vegetation Memory Models	95
A.4.4	Miscellaneous Figures	97
A.5	Declaration Of Authorship	100

List of Figures

1.1	Components of Vegetation Memory	4
1.2	BIEN PFT Coverage	6
1.3	Concept of Combining Plant Functional Trait Information with Species-Distribution Maps	7
1.4	Study Outline	8
2.1	Global Representation of the Normalised Difference Vegetation Index (NDVI)	10
2.2	Overview of soil moisture data	12
2.3	Overview of air temperature data	13
2.4	DEM elevation data	14
2.5	Global Vegetative Height (H) data as obtained via TRY	15
2.6	Global Leaf Nitrogen Content (N_{mass}) data as obtained via TRY	16
2.7	COMPADRE sites available to this study	17
2.8	GBIF <i>Plantae</i> Occurrence Overview	18
2.9	European Ombroptypes	19
2.10	South-American Ombroptypes	20
2.11	Australasian Ombroptypes	21
2.12	Kriging Concept	23
2.13	Vegetation Memory Model Flowchart	26
2.14	Memory Model NDVI Data Treatment	26
2.15	Memory Model Qsoil Data Treatment	27
2.16	Exploratory Vegetation Memory PCA	27
2.17	PCA Regression Coefficients	28
2.18	Variance Partitioning	29
3.1	Short	32
3.2	Short	33
3.3	Short	34
3.4	Short	35
3.5	Short	36
3.6	Short	37
3.7	Short	38
3.8	Short	39
A.1	Working Directories	55
A.2	HWSD slope aspect data	61
A.3	HWSD slope incline data	62
A.4	Data Overview (Iberian Region)	63

A.5	TRY Data (Iberian Region)	64
A.6	Data Overview (Caatinga, Brazil)	65
A.7	TRY Data (Caatinga, Brazil	66
A.8	Data Overview (Australia)	67
A.9	TRY Data (Australia)	68

List of Tables

1.1	Basic vocabulary of the ecological stability debate	1
1.2	Basic vocabulary of the vegetation memory framework	3
1.3	Basic vocabulary of Plant Functional Trait frameworks	5
2.1	Core Information about NDVI data	10
2.2	Core Information about ERA5 data	11
2.3	Core Information about HWSD data	14
2.4	Interpretation of Memory Model Coefficients	24
A.1	R Packages	54
A.2	HWSD Variables and Explanations	61

List of Abbreviations and Acronyms

H Vegetative Height

N_{mass} Leaf nitrogen content per leaf dry mass

π Period of Oscillation

ρ Damping Ratio

AIC Akaike Information Criterion

AVHRR Advanced Very High Resolution Radiometer

BIEN Botanical Information and Ecology Network

CHELSA Climatologies at High Resolution for the Earth's Land Surface Areas

COMPADRE COMPADRE Plant Matrix Data Base

CRU Climatic Research Unit

DEM Digital Elevation Model

DSSs Decision Support Systems

ECMW European Centre for Medium-range Weather Forecasts

ERA5 European Centre for Medium-range Weather Forecasts ReAnalysis 5

FDD Functional Diversity Dispersion

FDR Functional Diversity Richness

FSC Fast-Slow Continuum

FSC-1 Life History Speed within the Fast-Slow Continuum

FSC-2 Reproductive Strategy within the Fast-Slow Continuum

GBIF Global Biodiversity Information Facility

GIMMS Global Inventory Modelling and Mapping Studies

HWSD Harmonized World Soil Database

iDiv Deutsches Zentrum für integrative Biodiversitätsforschung

LCCSs Land-Cover Classification Systems

LHT Life History Trait

MODIS Moderate-Resolution Imaging Spectroradiometer

NDVI Normalized Difference Vegetation Index

PCA Principal Component Analysis

PFT Plant Functional Trait

Qsoil Soil Moisture

Qsoil1 Soil Moisture (0-7cm)

Qsoil2 Soil Moisture (7-28cm)

Qsoil3 Soil Moisture (28-100cm)

Qsoil4 Soil Moisture (100-255cm)

SPEI standardized precipitation evapotranspiration index

Tair Air Temperature (0-2m)

TRY TRY Plant Trait Data Base

VI Vegetation Index

1. Introduction

Ecosystems are subject to disturbance regimes which can undergo substantial alterations^[1] thus causing growing concern that future disturbance dynamics will exceed tipping points and cause irreversible shifts in ecosystem state in relation to **attractors**^[2] (see table 1.1 for definitions of bold font terms relating to the resilience debate). Such a **regime shift**^[3–5] can have a tremendous impact on local as well as global welfare of mankind^[6–8]. Therefore, policy makers have made it their mission statement to enforce actions which serve to maintain or boost **resilience**^[9]. Conceiving such strategies and evaluating their respective efficacies requires a thorough understanding of resilience theory and a robust set of tools to measure the resilience of ecosystems.

Generally, resilience can be understood as the capacity of a system to absorb **disturbances** and respond to changing conditions so as to still retain the same function, structure, identity and feedbacks^[5,10]. First introduced to biological sciences in 1973 by Holling et. al resilience thinking was envisaged to encompass two separate characteristics of ecosystems: *stability* and *resilience*^[10]. Stability was defined as the ability of a system to return to its original state after a temporary disturbance. The more stable a system, the less it fluctuates in its characteristics and the faster it returns to its initial conditions after a perturbation. Contemporary literature usually refers to this property as **engineering resilience**^[11–13]. Resilience, on the other hand, was characterised as the property of a system that allowed it to withstand disturbances without collapsing. The more resilient a system, the higher its chances of persisting under changing conditions. Today this is mostly referred to as **ecological resilience**^[14–16]. Within this early context on resilience thinking, a system could be resilient whilst not being very stable and fluctuating in its properties a great deal^[10].

Since then, the term ‘resilience’ itself has undergone a ‘greenwash’^[17], which has diminished its direct usefulness to researchers. This process has seen the term ‘resilience’ being introduced originally to ecological sciences but adapted by other branches of research (e.g. social sciences^[18]) and society (e.g. policy makers^[17]), modified for different purposes and re-introduced to ecological sciences with multiple altered definitions^[19].

Table 1.1: Basic vocabulary of the ecological stability debate - A few definitions that are key to the understanding of the resilience framework.

Term	Definition
Attractor	A regime towards which a system moves asymptotically ^[5,7,20–24] . Also known as <i>stable state</i> ^[25] , <i>(stable) equilibrium</i> ^[12] or <i>regime</i> ^[22] .
Disturbance	Any impact that perturbs a systems trajectory from a given attractor ^[15,22,26–28] .
Regime shift	The restructuring of a system from one set of dominant processes and feedbacks governed by one attractor to an alternate set governed by an alternate attractor ^[20,21,29–31] . Also known as <i>tipping point</i> ^[3] or <i>bifurcation</i> ^[5] .
Ecological Resilience	'[...] measures the magnitude of disturbance that can be absorbed before the ecosystem's structure changes' ^[32] . Also known as <i>General resilience</i> ^[24] . Ecological resilience is a measure of how much a system can be changed until it shifts from one stable state/attractor to another ^[10,15,16,19,24,32–34] .
Engineering Resilience	'[...] measures the speed of recovery after the disturbance' ^[32] . Also known as <i>Recovery</i> ^[15] , <i>Stability</i> ^[10] or <i>Resiliency</i> ^[24] . Engineering resilience is a measure of how fast a system reverts to its pre-change state. It can be measured by assessing the return time ^[10,15,16,19,22,24,32–34] .

Whilst the state of any given ecosystem can be assessed without entertaining the underlying implications of a resilience framework, the conclusions from such a study may grossly differ from the actual processes in nature. Assumptions on ecosystem functioning that are made without including resilience thinking, irrespective of this caveat, may result in particularly dire consequences if the systems in question (also referred to as focal systems^[21,35]) are production ecosystems^[19,21] (ecosystems which are used by humans for resource allocation).

However, simultaneous assessment of components of any given ecosystem is a difficult task that often defies completion^[15] with most studies only being able to focus on a select few components at a time. Consequently, recent studies have focussed on identifying and quantifying resilience characteristics of ecosystems^[36–39]. Hodgson et. al tackled the problem of resilience measurements by building a framework for resilience assessments by first representing resilience as two separate entities: (1) *Resistance* to perturbation and (2) *Recovery* from perturbation^[33]. Representing the *recovery* aspect contained within this framework to resilience is (amongst others) **ecological stress memory** (also known as ‘ecological memory’, see table 1.2 for definitions of bold font terms relating to the ecological memory framework). However, even quantifying this component of an entire ecosystem is a Sisyphean endeavour at best as ecological memory is a compound metric unifying a multitude of sub-processes^[40]. Hence, ecosystem-scale studies have refocussed their attention on identifying components of ecological stress memory such as **vegetation memory**^[1,40].

1.1 Vegetation Memory

Ecological memory effects can be understood as the influence of antecedent conditions on ecosystem properties given contemporary disturbance regimes^[11,41]. The existence of such temporal effects emphasize the necessity of considering time in ecological frameworks by providing a link between antecedent environmental conditions and plant performance^[36,40,42,43]. In fact, there is empirical evidence that past conditions inform contemporary vegetation morphology^[40], phenology^[11], primary productivity^[41,44], species composition^[40,45], and carbon cycles^[11,43].

Studying vegetation memory sheds light on an important set of ecosystem processes which contain an outstanding potential to make ecosystems resilient in the face of climate change^[40].

1.1.1 Terminology

Vegetation memory can - in the broadest of terms - be understood as:

The impacts of antecedent conditions on current vegetation productivity.^[11,32,40,41]

In a comprehensive paper, Ogle et al. expanded on this definition by identifying the following three important characteristics to vegetation memory^[41]:

1. *Length* - The extent of time through which past conditions significantly affect the current state of vegetation.
2. *Strength* - The magnitude of the effect that past events/conditions have on the current state of vegetation
3. *Temporal Patterns* - The variation in relative impacts of antecedent conditions at different points in time.

The natural world is inherently complex and so I find it useful to add another criterion:

4. *Extrinsic Source* - The nature of the antecedent extrinsic conditions that drive current states of vegetation (i.e. the origin/source of the extrinsic memory effect), including (but not limited to): herbivory pressure or climate events.

Tackling the characteristic of vegetation memory origin, Ogle et al. introduced the notion of **intrinsic** and **extrinsic** memory effects^[41] (see table 1.2).

Table 1.2: Basic vocabulary of the vegetation memory framework - A few definitions that are key to the understanding of vegetation memory and its components.

Term	Definition
Ecological Memory	Ecological memory is synonymous with information and material legacies: adaptations, individuals, and materials that persist after a disturbance and drive the responses to future disturbances ^[1,46] .
Vegetation Memory	Vegetation memory is defined as any response of a single plant or vegetation compounds after following disturbance event that alters the response of the plant/plant community towards future stress events. This includes a modification of interaction with other ecological components ^[40] .
Intrinsic Memory	Intrinsic memory refers to the influence of antecedent conditions of the focal system in determining contemporary conditions of the same system. Also known as <i>endogenous memory</i> . ^[41]
Extrinsic Memory	Extrinsic memory refers to the influence of antecedent conditions of the environment (usually abiotic, climate factors) in determining contemporary conditions of the focal system. Also known as <i>exogenous memory</i> . ^[41]

1.1.2 Ecological Relevance

Vegetation memory is identified as the effect of antecedent conditions on current vegetation patterns^[41]. The stronger and longer the memory effect, the more vegetation will be influenced by antecedent anomalies of the ecosystem and abiotic processes thus signalling higher sensitivity to these drivers^[32]. A higher sensitivity translates to lower resilience and so:

Vegetation memory has been proposed as a proxy for engineering resilience.^[11,47]

Due to the inverse relationship between vegetation memory and engineering resilience, patterns and characteristics of vegetation memory are a vital information criterion for Decision Support Systems (DSSs) aimed at conservation or management efforts.

1.1.3 Quantifying Vegetation Memory

Vegetation memory is influenced by a multitude of processes which range from abiotic characteristics^[32,36,43] to biotic system components^[1,32]. Comprehensive studies of vegetation response to biotic and abiotic forcing are numerous^[11,36,36,41,43,48]. These studies have established a multitude of approaches to delineating vegetation memory from a variety of data sets. Examples include the use of (1) correlation coefficients of different lags of drought indices^[11,49], (2) mixed effect models of antecedent vegetation and climate characteristics and subsequent model selection^[32], and (3) Bayesian models to distinguish intrinsic and extrinsic memory effects^[41]. Some common practices have emerged from this variety of approaches:

1. **Intrinsic Memory** is usually identified as the relationship between vegetation properties at time step t a prior time $t - 1$ (usually at monthly intervals)^[32]. See figure 1.1 for a visual representation.
2. **Extrinsic Memory** is attributed to combinations of temperature, precipitation, and/or drought records^[11,32,41,49]. See figure 1.1 for a visual representation.
3. **Model Selection** is used to determine the length of extrinsic memory effects by identifying the time window during which extrinsic forcing holds the most explanatory power of vegetation characteristics^[11,32,49].
4. **Anomalies** of both response and predictor variables are used as these are metrics which can be linked directly to perturbation events^[32].

Vegetation memory models can thus usually be summarised as:

$$Y_t = \alpha * Y_{t-1} + \sum_{e=1}^E (\beta_e * X_e) \quad (1.1)$$

with Y_t and Y_{t-1} denoting anomalies of vegetation properties at times t and $t - 1$ respectively, X_e indexing anomalies of the e 'th extrinsic force (out of a total E extrinsic drivers), and α and β representing the **coefficients of intrinsic** and **extrinsic** memory effects respectively. Rerunning such a model with independent, extrinsic inputs (X_e 's) at different time lags and subsequent model selection can then be employed to identify the length of the extrinsic vegetation memory related to the e 'th extrinsic force.

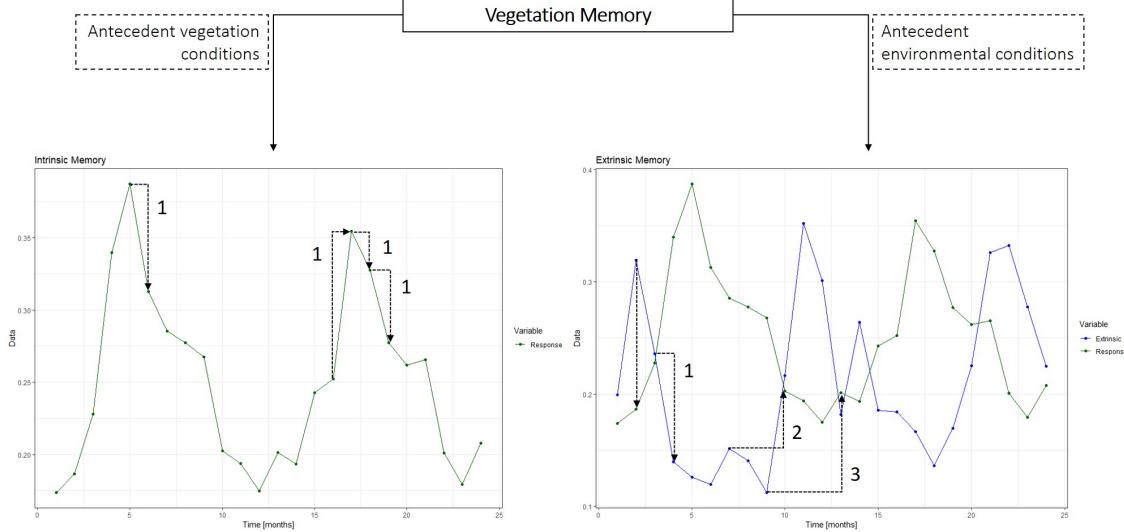


Figure 1.1: Components of Vegetation Memory - Conceptual layout of how intrinsic and extrinsic vegetation memory components can be identified from the data. The concept recognises two important variables: (1) *Response* - contemporary vegetation properties at time t , and (2) *Extrinsic* - antecedent environmental data. Within this framework, vegetation memory - both intrinsic and extrinsic - is identified as the predictability of the response variable through a regression onto itself and extrinsic variables respectively. Regressions of data onto each other are visualised as dashed arrows, lag effects are identified by numbers. Some parts of this figure have been generated via Chunk 27.

1.2 Plant Functional Traits

Studies of biotic forcing usually fall into one of two categories which can be seen as research representative of the world views of Schimper (focus on functional diversity) and Hubbel (focus on species diversity)^[50]. Whilst the latter represents traditional approaches to understanding the natural world, Schimpers framework has received increased attention over the past few years. A functional approach to ecosystem processes can potentially explain what functional expressions allow said processes to be maintained in their structure despite perturbations^[51–54]. One of the most prominent tools of these ‘Schimper’ studies is the *functional trait frameworks*. The most important of these frameworks when understanding vegetation processes - such as vegetation memory - is that of **Plant Functional Traits (PFTs)**.

Functional traits capture a multitude of characteristics without the necessity for a focus on phylogenies of study organisms^[55] thus making landscape-scale studies required for identifying vegetation memory patterns far more feasible. The diverse range in which any given **Plant Functional Trait (PFT)** (see table 1.3 for definitions of bold font terms relating to PFT frameworks) can be expressed are representative of the diverse strategies for establishing, growing and reproducing of plants^[56]. Hence, understanding how plant functional traits enable plants to persist despite (a)biotic forcing is a promising first step to understanding ecosystem-scale responses to climate change.

1.2.1 Terminology

Whilst PFTs have been linked individually to evolution^[57] and community composition patterns^[54,56], a change in their expression has been found to be correlated with climate change patterns^[58] thus suggesting a causal link between extrinsic vegetation memory and PFT expressions. In order to classify the way in which PFTs are related to community functions, Nock et al. present the concept of (1) **Effect Traits**, and (2) **Response Traits**^[55]. Classifying PFTs according to this framework can profoundly aid the understanding of their influence on spatio-temporal patterns of vegetation performance^[30,54].

Recent studies suggest that vegetation community functions are governed by integrated phenotypes, which can be regarded as specific combinations of individual PFT ranges, instead of single PFTs^[55,59] thus calling for a scientific approach focussed on **Functional Diversity Richness (FDR)** which can be identified via an approach in which PFT values of each individual are placed within a multidimensional system with each axis representing the range of values for one specific PFT^[60]. Just as with species oriented studies, a measure of richness alone may be insufficient to capture necessary information to understanding biological systems. Therefore, one may also want to use a measure of evenness which, in a functional biology setting, can be referred to as **Functional Diversity Dispersion (FDD)** which is representative of how evenly dispersed data records are in multidimensional space.

Recent studies in PFT research have focussed heavily on methods of dimensionality reduction by identifying the PFTs which capture a vast part of the global variation of plant function^[56,61]. Díaz et al. identified:

1. *Vegetative height (H)* an important criterion to accessing light resources
2. *Stem specific density (SSD)* reflecting a trade-off between growth potential and mortality risk
3. *Leaf area (LA)* vital consequences for leaf water balance
4. *Leaf mass per area (LMA)* a proxy for the trade-off between carbon gain and leaf longevity
5. *Leaf nitrogen content (Nmass)* reflects a trade-off between photosynthetic potential and acquiring nitrogen
6. *Diaspore mass (Dmass)* reflects a trade-off between seedling survival and colonization ability

as the most important PFTs to understanding the global spectrum of plant functioning^[61]. A similar framework has been established by Westoby & Wright in which three important PFT dimensions are recognised: (1) the leaf economic spectrum, (2) the seed size/mass spectrum, and (3) the height of the canopy at maturity^[50].

Table 1.3: Basic vocabulary of Plant Functional Trait frameworks - A few definitions that are key to the understanding of Plant Functional Trait frameworks.

Term	Definition
Plant Functional Traits (PFTs)	Plant Functional traits characterise morphological, biochemical, physiological, structural, phenological and/or behavioural aspects of organisms which influence the fitness of said organisms ^[55] .
Effect Traits	These determine a species influence on ecosystem processes ^[55] .
Response Traits	These determine a species ability to colonise a certain habitat and persist despite environmental pressures ^[55] .
Functional Diversity Richness (FDR)	Functional diversity richness measures the range of the PFT spectrum of a given study region/species assembly in multivariate trait space via a multivariate convex hull effectively measuring niche space ^[60] .
Functional Diversity Dispersion (FDD)	Functional diversity dispersion measure the dispersion of PFT values in multivariate trait space with respect to the local functional space centre effectively measuring niche segregation ^[60] .

1.2.2 Ecological Relevance

The positive effect of biodiversity on ecological stability has long been recorded^[51] and even led to the formulation of prominent concepts like Yacchi's insurance hypothesis which postulates that the loss of a subset of species can be compensated for by the presence of other species given a sufficient overlap in functional aspects of both thus leading to a stabilisation of ecosystem processes^[62].

PFTs capture information about a variety of ecosystem processes.^[63]

Most studies to date have focussed on species diversity^[5,6,19]. Some made use of the concept of functional species^[16] which presents researchers with a species classification system that is vague at best due to loose definitions of where to draw the lines between functional species. Functional trait biology, which captures the functional diversity of biological systems, is often employed as an improvement to the functional species/type concept^[40,42] as it eliminates the necessity of phylogenetic analyses^[55] and enables studies to capture intra-specific variation^[64–66].

PFTs can be captured irrespective of species relationships whilst presenting functional information at sub-species levels.^[64,65]

Thus, PFTs make for *highly informative* proxies of *ecosystem functioning* which can be recorded across *large spatial and temporal scales* since studies aren't constrained to a handful of species. Additionally, PFTs represent an improved understanding of functional expressions of plant communities by *expressing information at sub-species level*.

1.2.3 Unifying Information on Plant Functional Traits

PFTs are usually recorded through individual trait campaigns or for single-study purposes. Combining PFT information from these individual data sets can prove challenging due to different measurement practices and understandings of PFTs. A comprehensive handbook on PFT data collection practices by Pérez-Harguindeguy et al.^[67] presents an elegant solution to making PFT data sets more comparable. Additionally, PFT data sets have been aggregated into large data bases within online repositories such as:

- The **TRY Plant Trait Data Base (TRY)**^[66] is data base of PFT records connecting vegetation scientists across the globe and hosted by the Deutsches Zentrum für integrative Biodiversitätsforschung (iDiv). Within its current release version (version 4) TRY contains almost seven million PFT records for 1,800 traits with about half of all records being geo-referenced^[68].
- The **Botanical Information and Ecology Network (BIEN)**^[69] is a repository of global plant diversity, function, and distribution. Its current release (version 4.1) contains over 200 million records at global coverage^[70] (see figure 1.2).

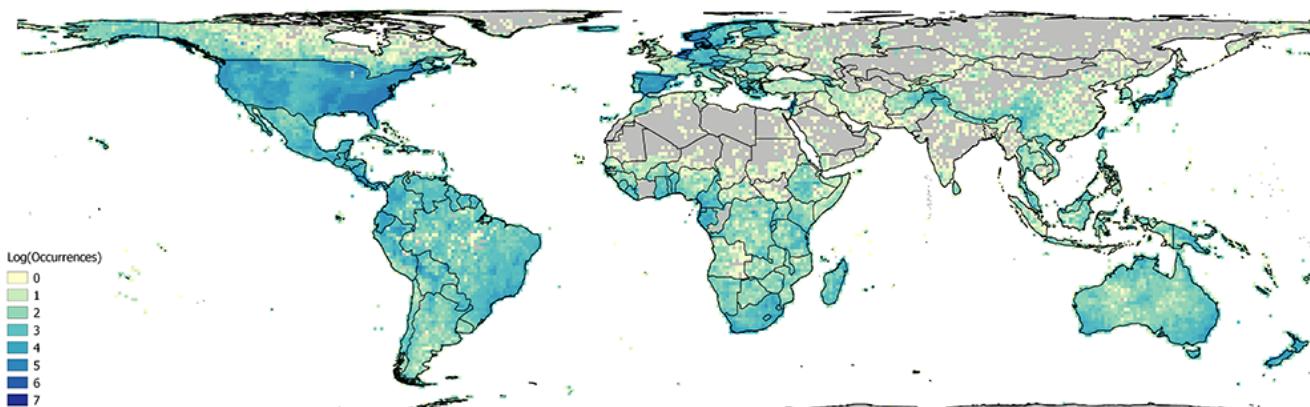


Figure 1.2: BIEN PFT Coverage - Geo-referenced records within version 4.1 of the BIEN data base at logarithmic scale^[70].

- The **COMPADRE Plant Matrix Data Base (COMPADRE)**^[71] contains Life History Trait (LHT), a subclass of PFTs pertaining to life history characteristics (e.g. longevity, reproductive mode, etc.), driven matrix population models which produce important information about community processes through time based on records of different LHTs^[72]. In its current release (version 4.0.1) it contains 7,024 matrix models^[73].

Combining information from these data bases should result in a holistic view of global plant function as such an approach would unify PFTs and LHTs to further a spatio-temporal understanding of global plant function.

1.2.4 Putting Plant Function on the Map

Vegetation Memory patterns are inherently spatial phenomena^[11,32,41,49]. So are patterns of plant functioning^[60]. Whilst the above mentioned data bases (TRY, BIEN, and COMPADRE) contain vast amounts of geo-referenced PFT and LHT records, almost no comprehensive, gapless, and internally consistent global products of plant function are available^[63,74].

Currently, the most commonly used method for extrapolating PFT records to local/global map products follows the methodology laid out by Ordonez & Svenning^[60] which has species-specific PFT means being combined with geo-referenced species occurrences (a conceptual depiction of this can be seen in figure 1.3).

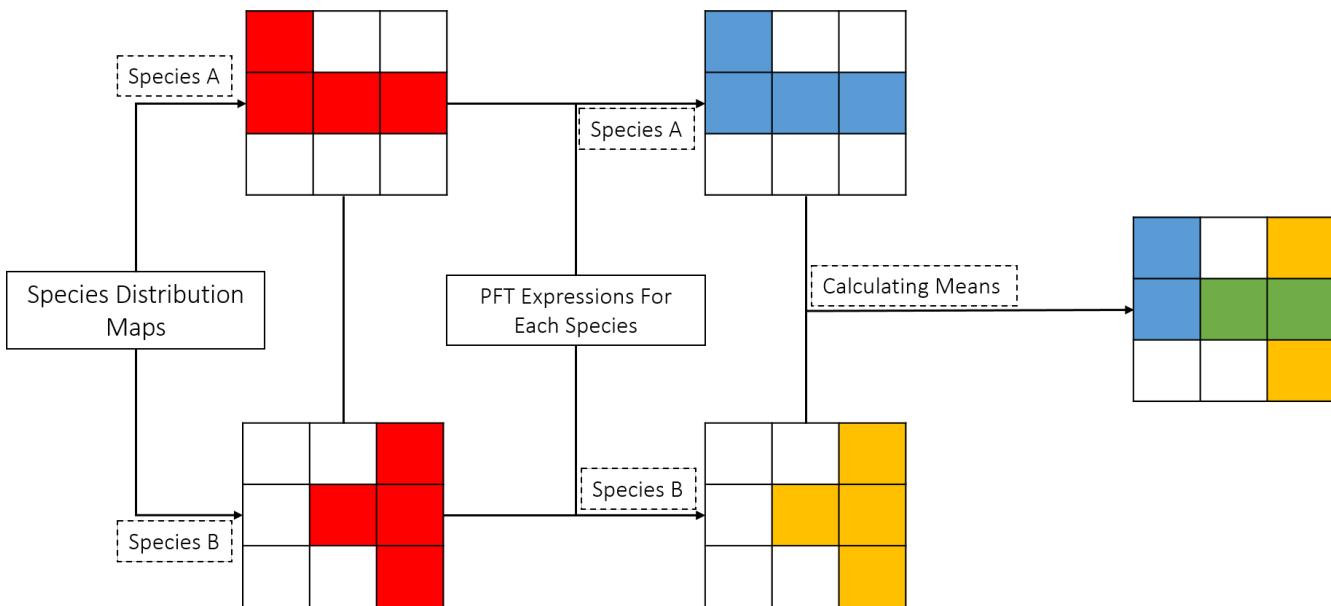


Figure 1.3: Concept of Combining Plant Functional Trait Information with Species-Distribution Maps - Conceptual framework of mapping plant function from PFTs and species distribution maps into gridded map products of mean PFT expressions. Inspired by Ordonez & Svenning^[60]. Red cells indicate presence of a species, blue and yellow cells depict the average PFT of species A and B respectively. Green cells represent the average of PFT means of species A and B whilst white cells denote the absence of species and thus PFT data.

Species occurrence data - referred to as **floral data** in the case of plant occurrences - can be obtained via the **Global Biodiversity Information Facility (GBIF)** data base which currently contains over 1 billion geo-referenced occurrence records of species across all kingdoms of life^[75].

Unfortunately, this approach is heavily reliant on geo-referenced occurrence records which are subject to sampling bias^[76] and renders intra-specific expressions of PFTs virtually non-existent as species are expressed via single mean values for each PFT. Zoological studies have resolved these issues by recording relative abundances of species in addition to occurrence records^[77].

Recent approaches in PFT ecology have seen the development of mapping procedures of PFTs according to abiotic conditions through linear models with spatial components^[74] and complex Bayesian methods^[63].

1.3 Thesis Outline

Remote sensing analyses have been the key to identifying vegetation memory patterns through time and space in a multitude of settings^[11,32,49]. Refining such approaches will facilitate important knowledge for DSSs on environmental processes in the face of climate change as well as changes in land use by mankind^[78]. Considering the direct link between vegetation memory and ecosystem resilience, this will garner an advanced comprehension of resilience and, in turn, enable maintenance and enhancement of ecosystems' resilience thus aiding to stabilize natural systems. It is therefore the **first main goal** (see figure 1.4 for an overview of how I am planning to achieve the research goals layed out here) of this study to

I. Identify vegetation memory patterns while improve on contemporary approaches.

To do so, I am going to answer the following research questions:

1. *Which variables (climate and biological) make up robust vegetation memory metrics?*
2. *How can we distinguish between intrinsic and extrinsic memory?*

A multitude of recent studies have identified vegetation memory characteristics of a diverse cast of focal ecosystems^[11,36,38,40,43,47,79,80]. However, only a few of these have also delineated the ecological processes and causal pathways which have led to the observed patterns of vegetation memory and hence resilience. One notable example of these research efforts is a study by Foley et al. which expanded on the drivers behind the ecosystem collapse of the Sahara and Sahel regions^[25]. Therefore, the **second main goal** of my thesis is to:

II. Determine how vegetation memory and plant function are linked.

Achieving this goal is possible by answering the following research questions:

1. *Which traits of biological function (PFT and LHT) are related to vegetation memory characteristics?*
2. *What biological traits cause areas to exert intrinsic and extrinsic memory?*

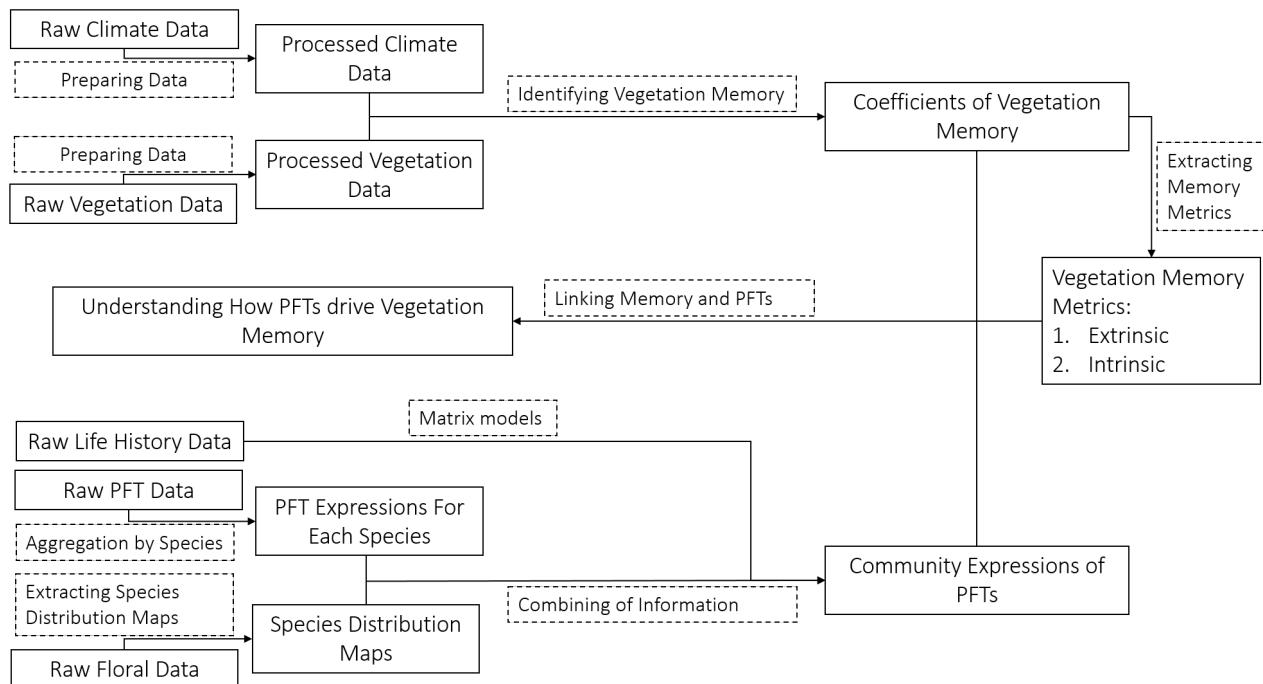


Figure 1.4: Study Outline - Conceptual study flow-chart. Data sets and products are shown in boxes with solid outlines whilst data handling and computational procedures are depicted via dashed-outline boxes.

2. Material & Methods

All analyses have been carried out in R^[81] using the functionality contained within the base installation as well as the packages lined out in table A.1. Chunk 1 contains the R commands used to install all packages needed to reproduce the analyses of this study from a base installation of R. Data sources and methods of data retrieval are stated for each data set individually.

2.1 Data

Identifying vegetation memory requires intrinsic data (e.g. proxies of vegetation performance), and extrinsic data (e.g. climate/weather indices)^[32,41]. Measures of plant function or life histories (i.e. PFTs and LHTs) of local vegetation may be useful in uncovering causal pathways leading to vegetation memory behaviour.

2.1.1 Vegetation Indices

Patterns of vegetation characteristics - both in time and space - can be assessed readily using remote sensing approaches^[36,47,82–86]. A Vegetation Index (VI) is a remote sensing proxy for different vegetation characteristics depending on the choice of VI.

2.1.1.1 Background

VI's have seen various applications in remote sensing including deriving vegetation dynamics on the scale of countries^[87–89] or regions^[85,90], and analysing resilience of vegetation assemblages to short-term climate anomalies^[32]. Biome classifications have undoubtedly been one of the most prominent usages of the VI's^[86,91–93] as have applications in agriculture^[94,95].

Whilst a vast arsenal of VI's are available to macroecologists (see Cammarano et al. for an overview^[85]), one of the main considerations in choosing an appropriate VI is that of **spatial resolution versus length of time series**. As satellite-born sensors receive hardware updates or other improvements - e.g. Advanced Very High Resolution Radiometer (AVHRR) versus Moderate-Resolution Imaging Spectroradiometer (MODIS) sensors - , spatial resolution capabilities of VI's increase. However, we can't enhance the resolution of older records of VI's and are thus limited to lower resolution VI's if longer time series are desired. Since vegetation memory is a temporal phenomenon first and foremost, I have prioritised **time series length** over spatial resolution. Within the confines of my thesis, VI's represent the intrinsic characteristics of vegetation systems which will both be used as response and explanatory variables as outlined in figure 1.1.

2.1.1.2 Normalised Difference Vegetation Index (NDVI)

The Normalized Difference Vegetation Index (NDVI) has been selected as a proxy of vegetation characteristics in this study due to (1) its nature as an information criterion of biomass^[96] and vegetation cover^[85] (indicators of vegetation performance and composition), (2) the availability of a long time series data with global coverage^[97], and (3) its demonstrated utility in various ecosystem studies^[84] including assessments of vegetation sensitivity and memory^[32,36,49]. See table 2.1 for an overview of the core characteristics of the NDVI data set.

Rouse et. al^[98] first introduced the NDVI in 1974 and initially coined it 'Vegetation Index'. The name was subsequently changed to NDVI when other VI's were proposed in the ensuing years alongside the advent of remote sensing studies which relied on VI's^[82,99,100].

The NDVI is a composite VI that factors in measurements in the near infra-red wave band (NIR, 0.58-0.68 μm) and the red wave band (RED, 0.75-1.10 μm). These bands belong to the spectrum of light that is absorbed by chlorophyll and thus provide information on green vegetation^[101]. The formula for NDVI is as follows^[87,101]:

$$NDVI = \frac{\rho_{\text{NIR}} - \rho_{\text{RED}}}{\rho_{\text{NIR}} + \rho_{\text{RED}}} \quad (2.1)$$

with ρ_{NIR} being the reflection in the NIR band and ρ_{RED} being the reflection in the RED band. The NDVI is bound between 0 and 1 with higher values representing denser vegetation. For a more detailed mathematical background on the calculation of NDVI scores see Matsushita et. al^[99].

NDVI data sets are readily available via the Global Inventory Modelling and Mapping Studies (GIMMS). For this study the GIMMS3g data set was used (available at <https://ecocast.arc.nasa.gov/data/pub/gimms/3g.v1/>), which provides AVHRR NDVI data as 15-day maximum-value composites from 1982 to 2015 at $0.083^\circ \times 0.083^\circ$ resolution^[102]. See figure 2.1 for a representation of one such global monthly composite. The data was processed in R, downloaded and compiled into monthly composites using the raster^[103] and **gimms** package^[104]. Chunk 8 contains the R-code used to process the data.

Table 2.1: Core Information about NDVI data - Characteristics of the GIMMS NDVI3g data set (v.1).

Characteristic	Data
<i>Resolution</i>	$0.083^\circ \times 0.083^\circ \sim 9.27\text{km} \times 9.27\text{km}$
<i>Time Series</i>	Bi-weekly intervals from January 1982 to December 2015
<i>Source</i>	https://ecocast.arc.nasa.gov/data/pub/gimms/3g.v1/

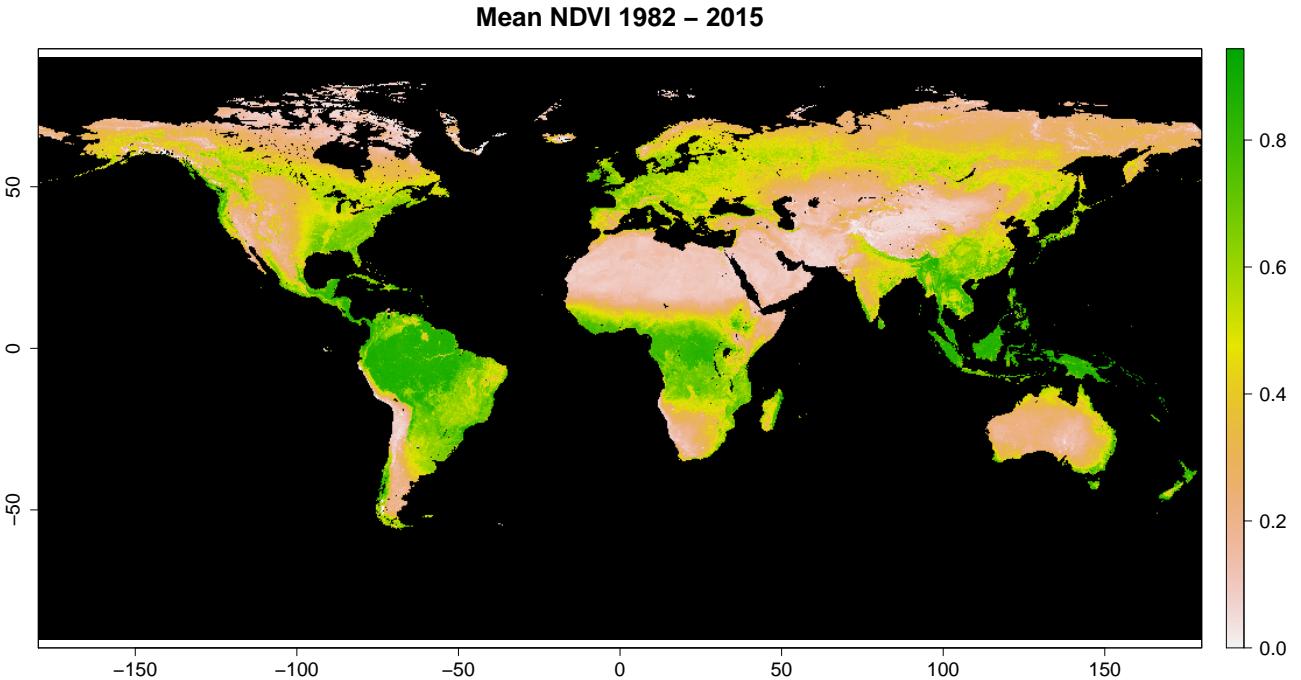


Figure 2.1: Global Representation of the Normalised Difference Vegetation Index (NDVI) - Mean of monthly maximum composite representations of NDVI from January 1982 to December 2015 at global coverage. Higher values of NDVI indicate higher vegetation coverage, plant biomass, and primary production. Figure established via Chunk 14.

2.1.2 Abiotic Data

2.1.2.1 European Centre for Medium-range Weather Forecasts ReAnalysis 5 (ERA5)

Indices of climate metrics are used for the calculation of extrinsic memory components as outlined in figure 1.1. Contemporary studies of vegetation memory largely rely on *precipitation* or *drought* data obtained via the standardized precipitation evapotranspiration index (SPEI) drought index^[32,47], and the Climatic Research Unit (CRU) data set^[11]. These data sets are *observational* data sets as they are built from observational data often involving simple interpolation methods^[105].

My study is using the **European Centre for Medium-range Weather Forecasts ReAnalysis 5 (ERA5)**^[106] data set which is a **reanalysis** data set. Reanalysis data sets are *self-consistent, gap-less* (in time and space) data sets and thus **superior** to many observational data sets. The superiority of ERA5 over frequently used climate data sets is largely due to:

1. The **volume of observational data** used to create the ERA5 product with data types ranging from satellite data to weather station/independent institute data collection efforts, and station data from a wide variety of data providers^[107]. Traditional, observational data sets are often characterised by their individual biases in sampling, and coverage representativeness.
2. Its **sophisticated nature** building upon data assimilation procedures^[108], and complex models^[109] to shed light on physical processes^[110]. As such, ERA5 is currently the *state-of-the-art* climate reanalysis product in climate science and benefits from the developments in data assimilation methodology, understanding of physics of the climate system and their interactions, which have come about in recent decades.

Finally, ERA5 improves on other prominent climate data sets in ecological studies like the aforementioned CRU and SPEI, WorldClim^[36], or the Climatologies at High Resolution for the Earth's Land Surface Areas (CHELSA) data set^[74] in a multitude of ways since it offers:

1. Superior *Temporal Resolution* (superior resolving of climate extremes)
2. Superior *Spatial Resolution* (advanced resolving of local/topographical features)

See table 2.2 for an overview of core characteristics of the ERA5 data set.

Table 2.2: Core Information about ERA5 data - Characteristics of the ERA5 data set.

Characteristic	Data
<i>Resolution</i>	$\sim 30\text{km} \times 30\text{km}$
<i>Time Series</i>	Hourly intervals from January 1950 to TODAY
<i>Source</i>	https://apps.ecmwf.int/data-catalogues/era5

Extrinsic memory effects have largely been understood through processes of drought and water limitation^[11,40,42,44,111]. Hence a large number of contemporary studies of vegetation memory have included some form of information criterion on water availability, through precipitation records^[36,43], or drought indices^[11,32,47]. Additionally, temperature has been found to be a major driver of drought severity, plant morphology, and vegetation memory effects^[36,43,45].

ERA5 data is implemented into my analyses of vegetation memory using two environmental characteristics: (1) Soil Moisture (in four layers), and (2) Air Temperature (as observable at a height of 2m above the ground).

See Chunk 4, Chunk 5, and Chunk 6 for the codes used to download ERA5 data from the European Centre for Medium-range Weather Forecasts (ECMWF) servers, aggregate data to full time series and fix gridding mismatches to be comparable to GIMMS data.

Soil Moisture recorded in $\frac{m^3_{H_2O}}{m^3_{soil}}$

My study uses *Soil Moisture* (*Qsoil*) as a proxy of local water regimes. Precipitation events are subject to further soil processes such as pore connectivity for precipitation water to be available to plants^[112]. Soil moisture should thus serve as a much more direct proxy of local water regimes. Additionally, ERA5 includes four distinct layers in the soil for the calculation of *Qsoil* indices (see figure 2.2 for an overview): (1) Soil Moisture (0-7cm) (*Qsoil1*), (2) Soil Moisture (7-28cm) (*Qsoil2*), (3) Soil Moisture (28-100cm) (*Qsoil3*), and (4) Soil Moisture (100-255cm) (*Qsoil4*). Typical drought indices (i.e. SPEI) do not allow for this additional distinction.

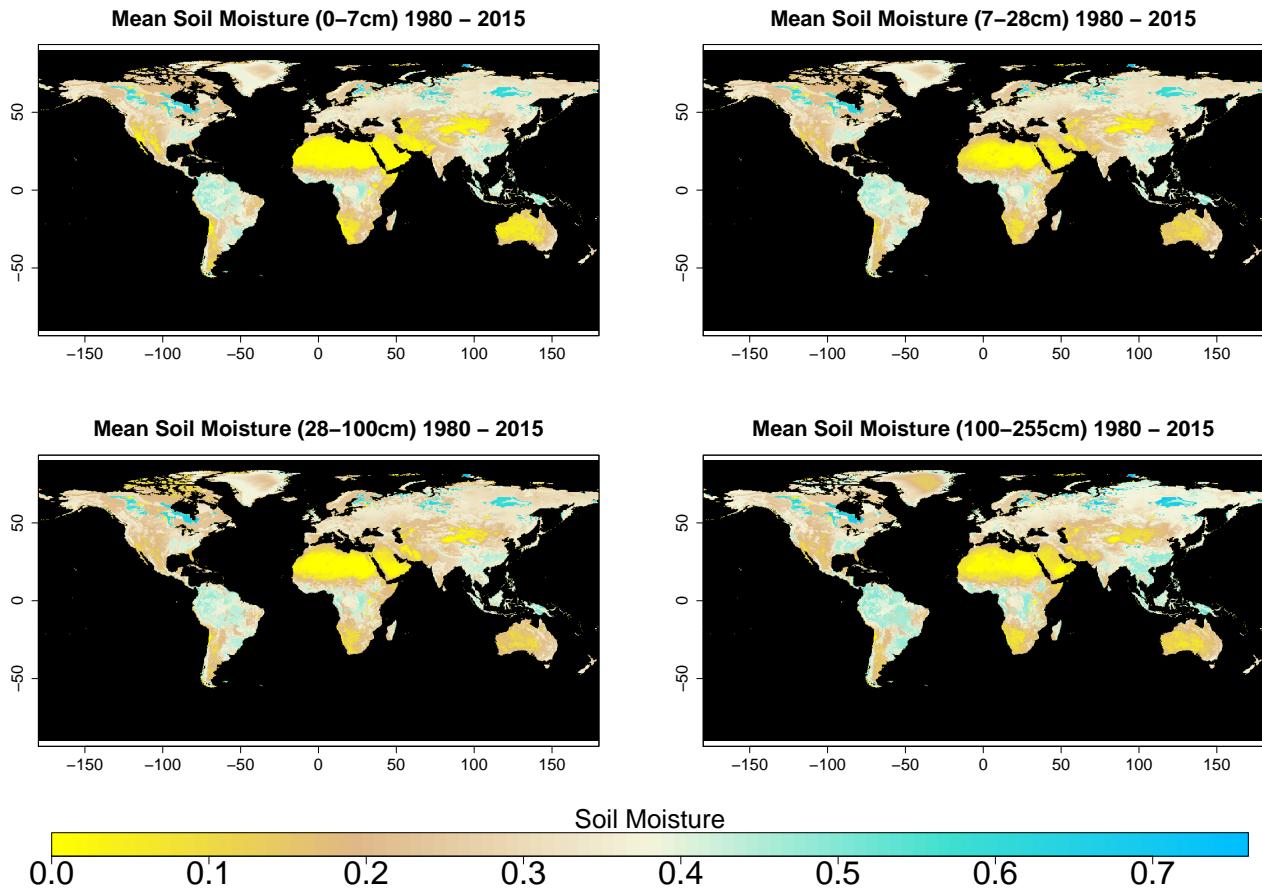


Figure 2.2: Overview of soil moisture data - Soil moisture data presented in layers as obtainable via the ERA5 reanalysis product (Air Temperature (0-2m) (*Tair*), *Qsoil1*, *Qsoil2*, *Qsoil3*, *Qsoil4*). Figure established via Chunk 15.

Within ERA5, unfrozen ground water (θ) across all four soil layers (k) is defined as:

$$\bar{\theta} = \sum_{k=1}^4 (R_k * \max[f_{liq;k}\theta_k, \theta_{pwp}]) \quad (2.2)$$

with R_k being the root fraction of soil layer k which is a fixed metric according to Land-Cover Classification Systems (LCCSs), and the statement $\max[f_{liq;k}\theta_k, \theta_{pwp}]$ calculating the amount of unfrozen soil water in soil layer k . $f_{liq;k}$ is a parametrised function of soil temperature; θ_{pwp} denotes the permanent wilting point according to soil texture. For a more in-depth explanation of how *Qsoil* is calculated within ERA5, see the IFS Documentation CY45R1 Chapter 4 *Physical Processes*^[110].

Air Temperature recorded in *K*

Temperature indices have been linked successfully to vegetation sensitivity^[36], tree-ring growth^[45,113], global primary production^[83], as well as severe drought events with possibly devastating consequences to local vegetation^[114].

ERA5 recognises several temperature variables (e.g. soil, snow and air temperature)^[110]. Within this study, I use Tair as contained within the ERA5 data set, due to the demonstrated impact of Tair on different aspects of plant physiology and plant morphology which may manifest in vegetation memory effects. See figure 2.3 for an overview of the ERA5 Tair data.

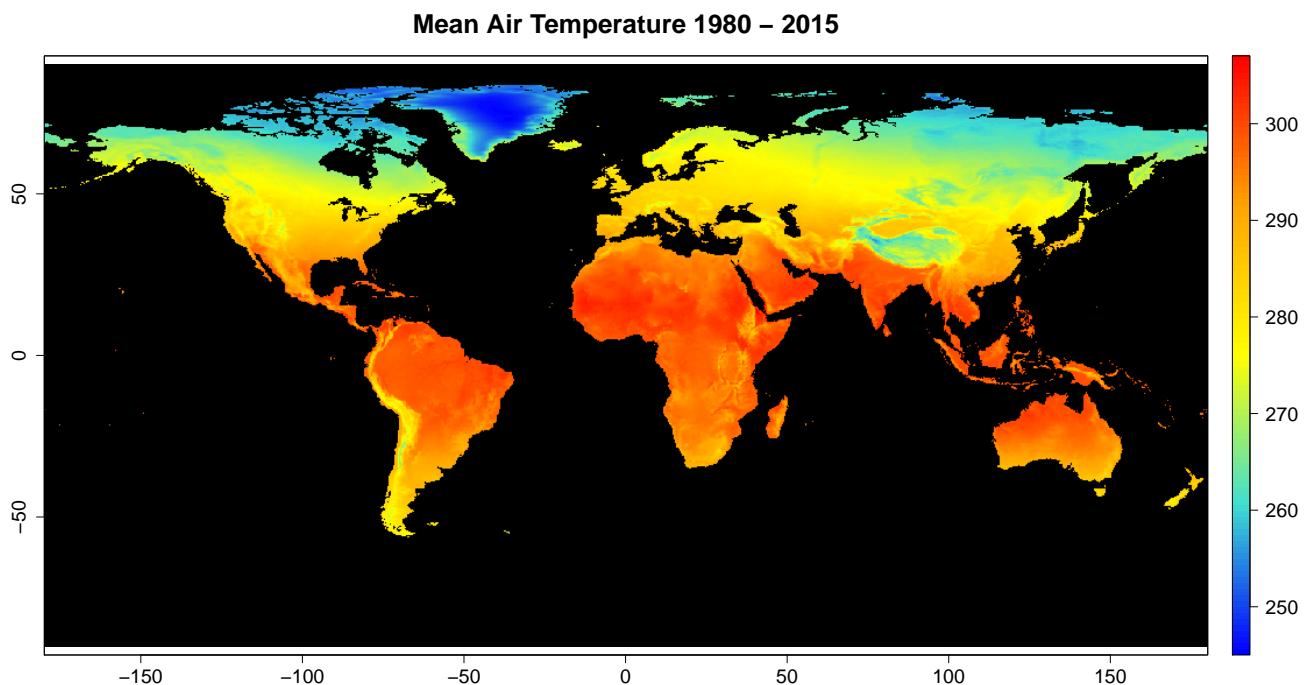


Figure 2.3: Overview of air temperature data - Air temperature data presented in as obtainable via the ERA5 reanalysis product (Tair). Figure established via Chunk 16.

2.1.2.2 Digital Elevation Models

Further abiotic data is needed for data preparation as outlined in section 2.3.1.1 To this end, I am using Digital Elevation Model (DEM) data obtained via the Harmonized World Soil Database (HWSD) which contains different DEM outputs at 3 and 30 arc-second resolution^[115,116]. See table 2.3 for an overview of HWSD DEM data characteristics.

Table 2.3: Core Information about HWSD data - Characteristics of the HWSD data set.

Characteristic	Data
<i>Resolution</i>	$3\text{arc} \times 3\text{arc} \sim 31m \times 31m$
<i>Time Series</i>	Static
<i>Source</i>	http://www.fao.org/soils-portal/soil-survey/soil-maps-and-databases/harmonized-world-soil-database-v12/en/

For an overview of all variables from the HWSD used within this study see table A.2. See figure 2.4 for a global representation of HWSD DEM data and figure A.2 and A.3 for an overview of HWSD slope aspect and incline data respectively as described in table A.2. These haven been rescaled to match GIMMS resolution (see table 2.1).

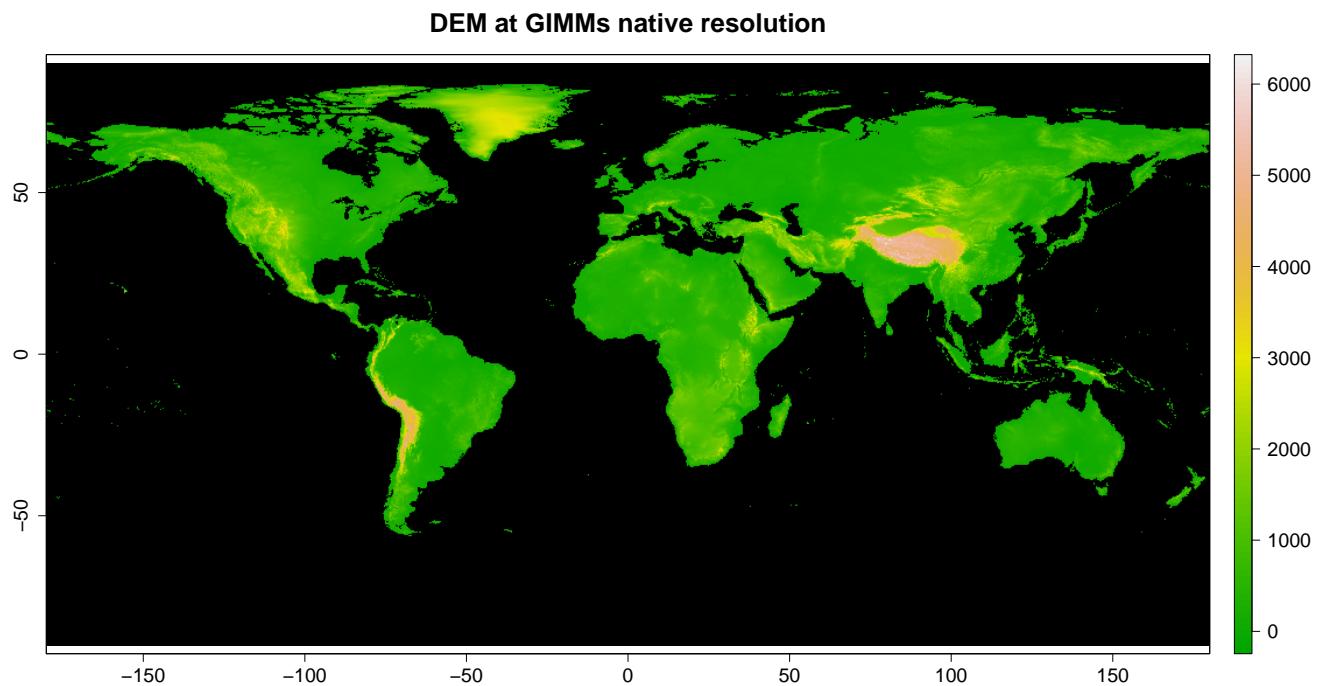


Figure 2.4: DEM elevation data - HWSD elevation data at GIMMS resolution. Figure established via Chunk 17.

2.1.3 Plant Functional Trait Data

Contemporary studies of vegetation memory largely describe memory effects in terms of spatial patterns and effect sizes, but forego analyses of plant physiology or morphology to **explain causal pathways** leading to vegetation memory.

Expressions of plant morphology and physiology are manifold and so one may wish to enlist multiple different proxies of these biological properties to rationalise vegetation memory effects.

2.1.3.1 TRY

Within this thesis, I am relating vegetation memory effects to two different PFTs which have been selected because they represent two important plant characteristics/trade-off axes in plant function (1) Vegetative Height, and (2) Leaf nitrogen content per leaf dry mass.

Additionally, these two PFTs correspond to two of the three PFT domains (i.e. stems and leaves) layed out in Westoby & Wright's PFT framework^[50]. Examining PFTs of multiple aspects of plant function in the face of adverse events is especially important as plant performance levels during and after disturbances often incur responses of all aspects of individual plant functional domains^[59].

For visualisations of the raw data, see figures 2.5 and 2.6. These plots clearly depict the limitation of TRY (and other PFT data bases) - sampling bias and effort^[76]. Certain regions of the earth are well-sampled for PFTs (e.g. the Pyrenees) although these sampling schemes are often limited to plot-level sampling campaigns. This data limitation highlights the potential importance of a mapping approach as lined out in figure 1.3. The TRY data used for my thesis has been obtained via TRY on 07/08/2018.

1. **Vegetative Height (H)** recorded in cm . Taller plants are better at pre-empting light resources and dispersing diaspores^[61].

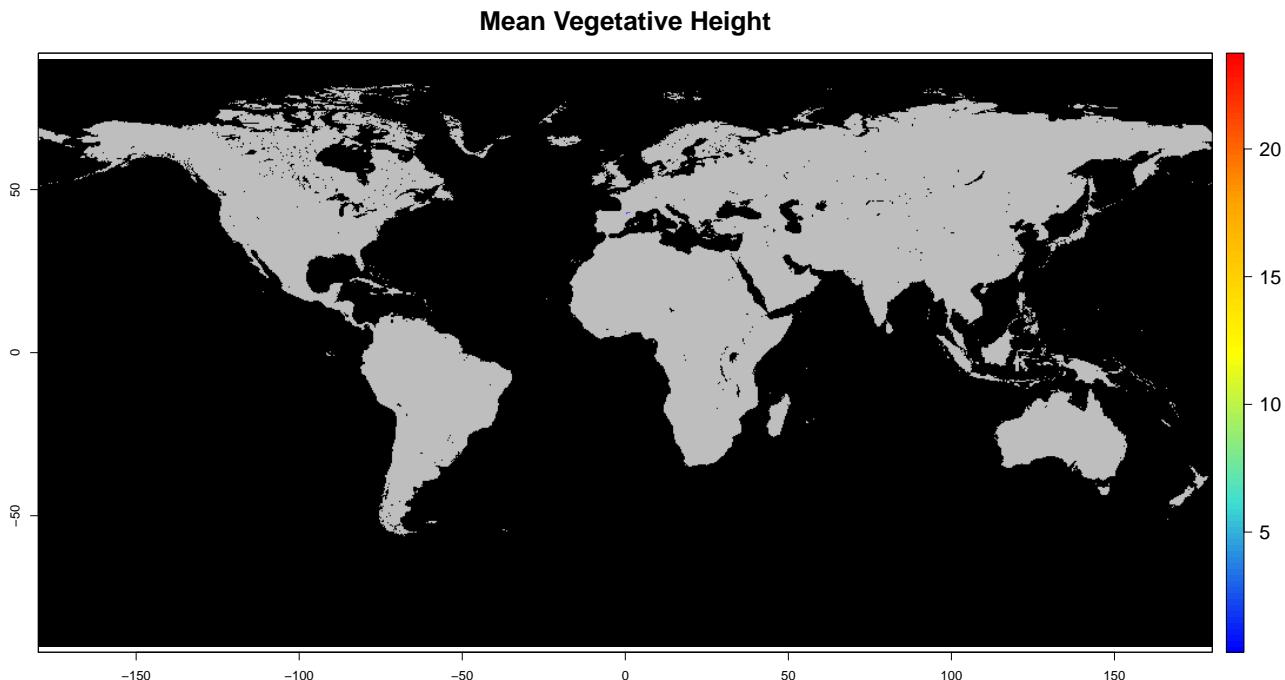


Figure 2.5: Global Vegetative Height (H) data as obtained via TRY - H data obtained via TRY, aggregated to rasters at GIMMS resolution. Grey pixels indicate terrestrial regions for which no H data is available. Figure established via Chunk 20.

3. **Leaf nitrogen content per leaf dry mass (N_{mass})** recorded in $\frac{mg_N}{g_{leaf}}$. This is a valuable proxy for different strategies of resource allocation to leaf longevity vs. photosynthetic potential^[61].

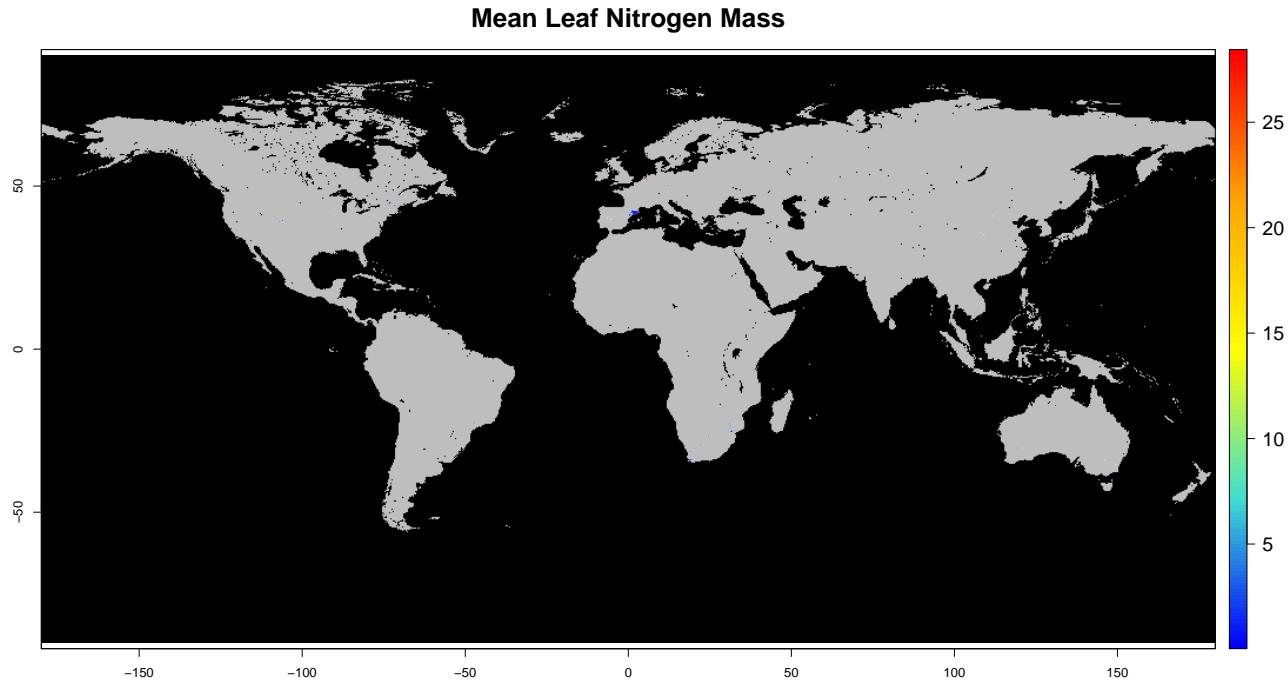


Figure 2.6: Global Leaf Nitrogen Content (N_{mass}) data as obtained via TRY - N_{mass} data obtained via TRY, aggregated to rasters at GIMMS resolution. Grey pixels indicate terrestrial regions for which no N_{mass} data is available. Figure established via Chunk 21.

2.1.3.2 COMPADRE

PFTs convey information as *snapshots* of reality as their data sets usually do not contain repeated measures of the same individuals over a period of time. Another set of metrics related to plant characteristics through time are **LHTs**. As such LHTs capture different information than PFTs and should thus be regarded as *complementary* to PFTs.

LHT information for plants and animals can be retrieved via COMPADRE; an extensive data base of different species and plot-scale ecosystems at different locations around the Earth^[71]. COMPADRE is built around observational data and matrix models^[117] which can be used to extract valuable information about temporal processes in biological communities^[118,119]. Recent research based upon this data base has revealed robust trade-off axes in plant^[72,120] as well as animal life histories^[121,122]. These axes are usually referred to as the **Fast-Slow Continuum (FSC)**^[72,120,122,123].

Information about LHTs has been demonstrated to be of use in trying to understand biological responses to temporal phenomena^[123] and should thus prove crucial in delineating causal, biological pathways to establishing vegetation memory. The outputs of COMPADRE matrix models are manifold^[120] offering access to a host of biologically relevant proxies of population processes. For this study, I have selected the following COMPADRE outputs as the most likely to be related to vegetation memory characteristics:

1. The **Fast-Slow Continuum (FSC)** is a Principal Component Analysis (PCA)-based approach whose two main axes capture over 60% of the variation in plant life history strategies^[120]. I hypothesize that expressions of the FSC will strongly interact with vegetation memory effects due to the holistic nature of the FSC and the temporal nature of both LHTs and vegetation memory effects. The two axes read as follows:

- (a) *Life History Speed within the Fast-Slow Continuum (FSC-1)* explains around 35% of global LHT variation and contrasts species of low generation times, early maturity, and fast growth with species of high generation times, delayed maturity, and slow growth^[120]. Fast species rank low on this axis^[72].
 - (b) *Reproductive Strategy within the Fast-Slow Continuum (FSC-2)* captures approximately 25% of global LHT variation and contrasts species low reproductive output with species that reproduce much, often, and for a long time^[120]. Species of low reproductive output rank low on this axis^[72].
2. **Reactivity** is a *first time-step* information criterion and a proxy for instantaneous biological responses^[124] and may thus be an important criterion in areas of high vegetation memory and thus low engineering resilience.
 3. The **Damping Ratio (ρ)** can be regarded as a measure of intrinsic resilience as it indicates how fast transient dynamics of a perturbation event decay; the larger ρ , the faster the population will converge^[124].
 4. The **Period of Oscillation (π)** is a measure of population life cycles in periodic environments with more rapidly growing populations usually being classified by higher π values^[125].

With COMPADRE, data for all variables contained within COMPADRE can be retrieved for all locations contained within the current release of the data base. For an overview of the COMPADRE sites available to this study using COMPADRE 4.0.1, see figure 2.7.

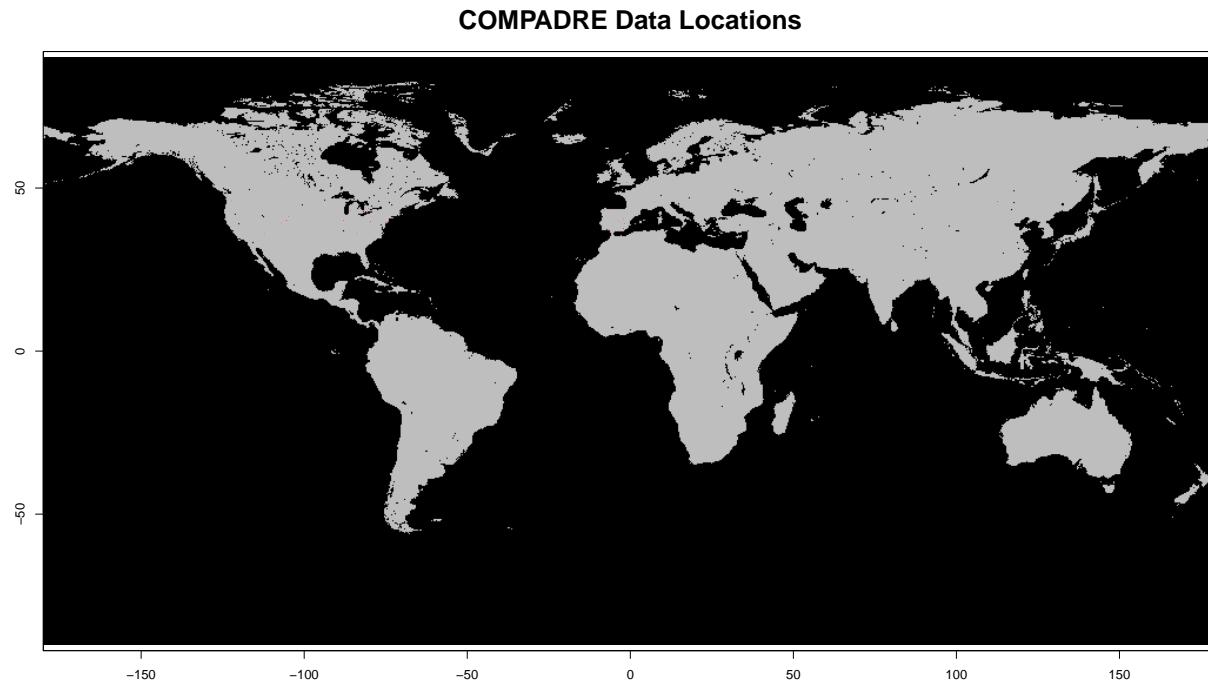


Figure 2.7: COMPADRE sites available to this study - Data availability within COMPADRE 4.0.1 aggregated to a raster of GIMMS resolution. Presence of data per grid cell is indicated with red shading. . Figure established via Chunk 22

Like with PFTs, sampling effort for LHTs within the COMPADRE scheme can only cover a limited range of geological locations. The best sampled for regions on Earth - in terms of COMPADRE sites - are the Iberian Peninsula and the contiguous United States of America. To date, no mapping approach beyond the rasterising of plot-level data of COMPADRE LHTs has been proposed.

2.1.4 Floral Data

Extrapolating PFT records from species-referenced records to spatial products using species-specific PFT means and species occurrence records as layed out by Ordonoez & Svenning^[60] (see figure 1.3) may prove to overcome the data limitations of TRY PFT data (see figures 2.5 and 2.6).

Plant species occurrence records can be obtained via floral data repositories such as GBIF (see figure 2.8 for a representation of GBIF plant occurrence data). Data in GBIF are stored as geo-referenced records of species presence. Generating maps from geo-referenced data points can be achieved by aggregating data points to rasters of a desired extent and resolution (i.e. GIMMS resolution).

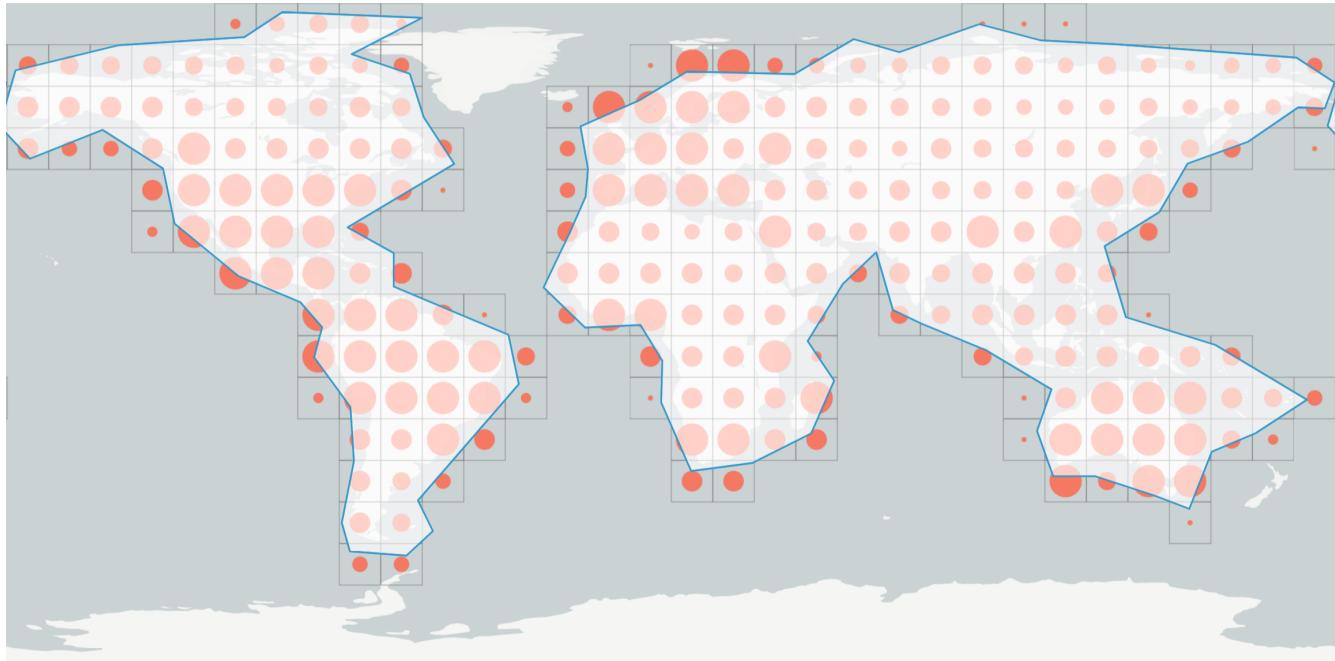


Figure 2.8: GBIF *Plantae* Occurrence Overview - Global representation of occurrence records for species within the *plantae* kingdom available via GBIF^[75] as of 21/04/19. Figure generated using the GBIF occurrence data exploration tool^[126] obtaining records from 1982 to 2015 (same as GIMMS data availability). Polygons have been drawn to omit occurrence records of marine plantae species. Larger red plotting symbols indicate greater amounts of geo-referenced occurrence records.

The floral data used within this study have been obtained for the time period of 1982 to 2015 (the same time span as the data availability for GIMMS NDVI data) via GBIF on 19/04/19 through to 21/04/19.

2.2 Study Regions

The spatial extent of study sites for assessing vegetation memory effects was limited to regions which were (1) suitable for remote sensing studies, (2) relatively well-sampled in terms of PFTs, LHTs and floral data, and (3) contain large areas classified as **drylands**. Since vegetation memory is largely researched in terms of water limitations and precipitation events, dryland regions (as classified via ombrotypes within Rivas-Martínez et al. [127]) are expected to allow for identification of extrinsic memory effects.

Within R, limiting to study regions is done using shapefiles (<http://www.naturalearthdata.com/downloads/10m-cultural-vectors/>) and the `rgdal`^[128] package (see Chunk 3).

2.2.1 Iberian Region

The Iberian region - said to encompass Portugal, Spain, Andorra and France within this study - has been selected as a study region due to a clear gradient in ombrotypes (see figure 2.9), its high density of TRY PFT and COMPADRE LHT data availability, and previously reportedly clear patterns of vegetation memory effects^[32] and vegetation sensitivity^[36].

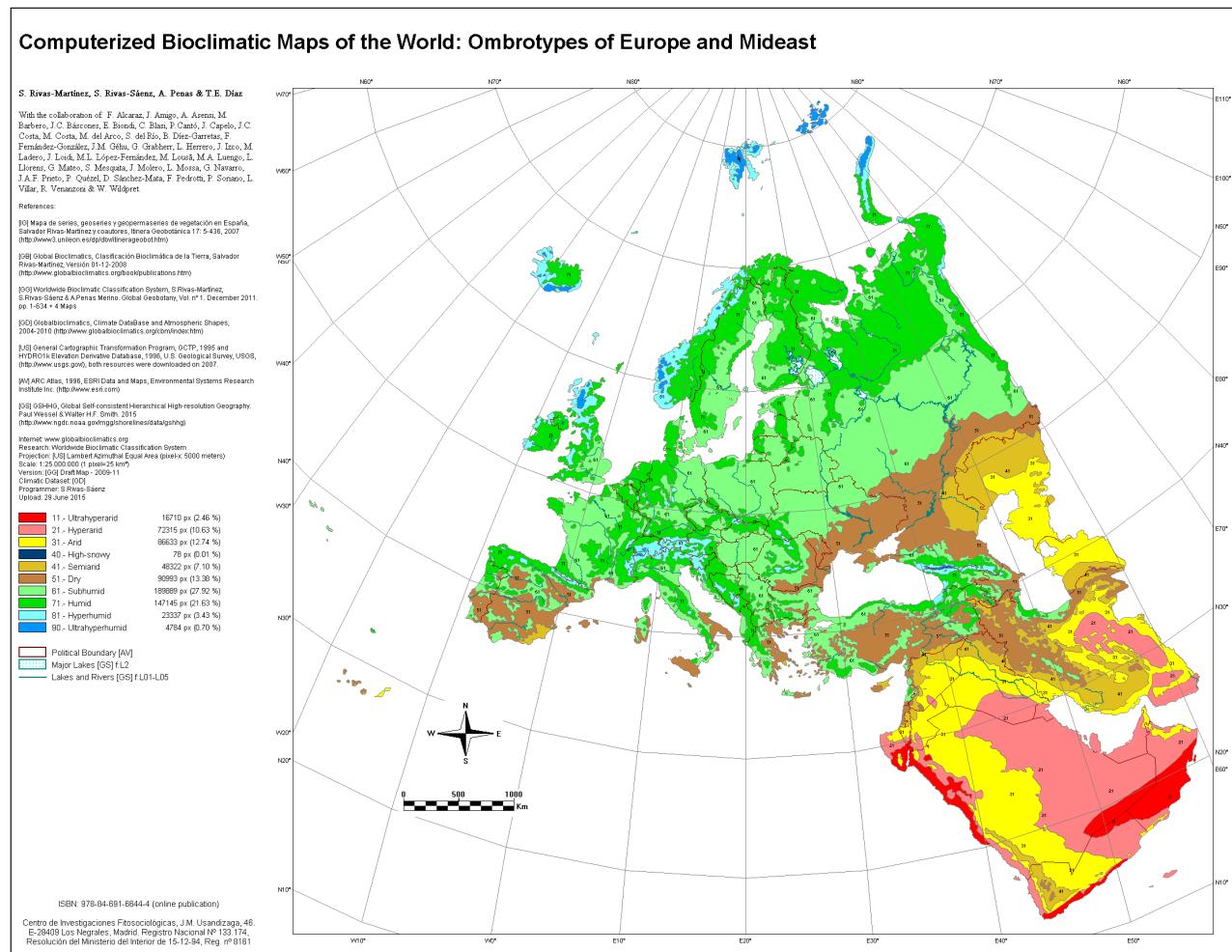


Figure 2.9: European Ombrotypes - Regions of mean precipitation rates of biological relevance across Europe according to Rivas-Martínez et al. [127] and retrieved via the Worldwide Bioclimatic Classification System^[129].

See figures A.4 and A.5 for an overview of GIMMS NDVI, ERA5 data, and TRY PFT data (both raw and extrapolated) across the Iberian region.

2.2.2 Caatinga, Brazil

The Caatinga in Brazil - a dryland region located within northeastern Brazil - has been selected as a study region due to a predominance in dry ombrotypes and gradients of ombrotypes at its edges (see figure 2.10) as well as previously identified strong vegetation memory effects^[32].

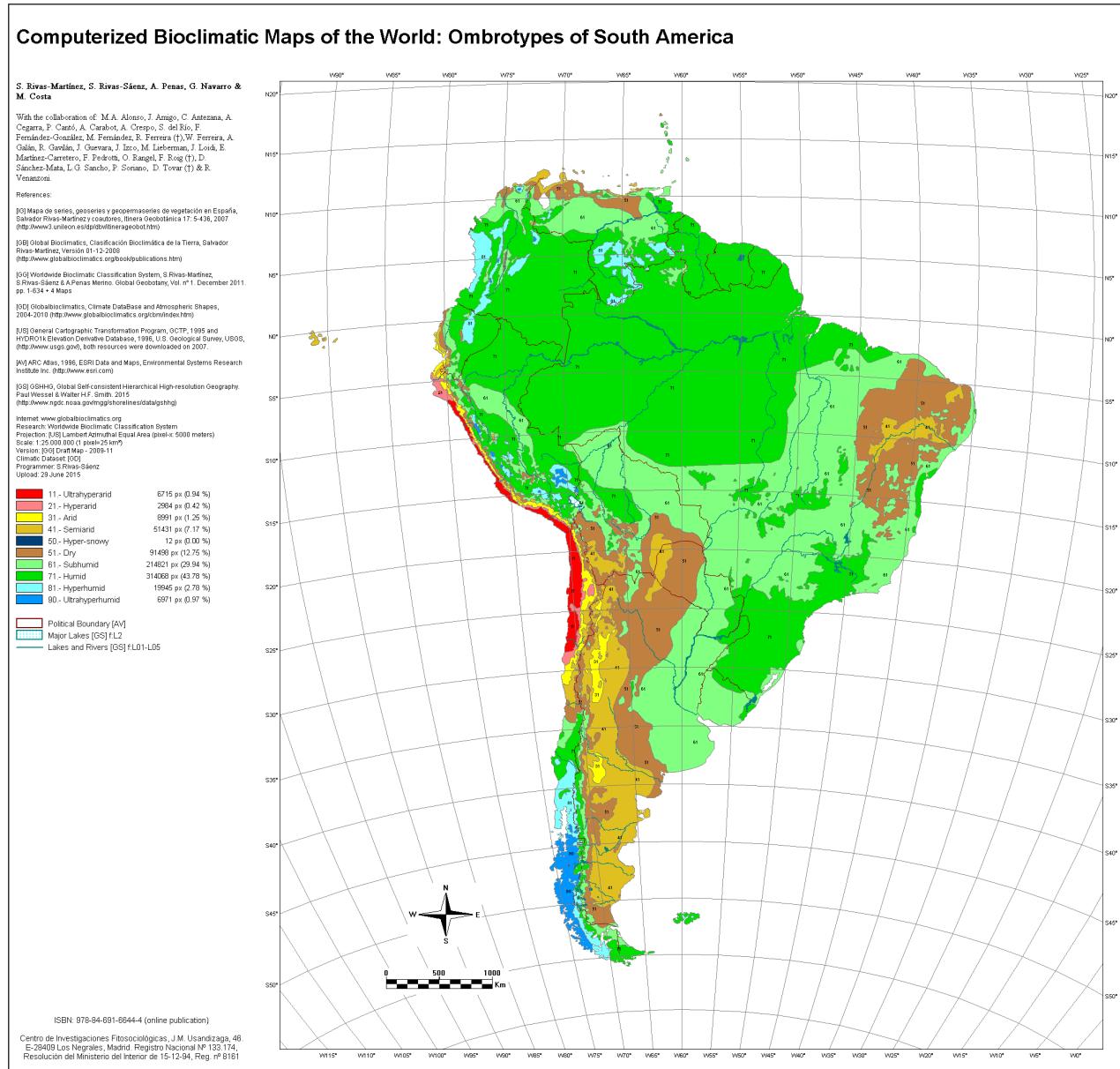


Figure 2.10: South-American Ombrotypes - Regions of mean precipitation rates of biological relevance across South-America according to Rivas-Martínez et al.^[127] and retrieved via the Worldwide Bioclimatic Classification System^[130].

See figures A.6 and A.7 for an overview of GIMMS NDVI, ERA5 data, and TRY PFT data (both raw and extrapolated) across the Caatinga.

2.2.3 Australia

Australia has been selected as a study region due to a predominance in dry and even arid ombrotypes as well as gradients of ombrotypes on its eastern coast (see figure 2.11). Previous studies have identified clear patterns of vegetation memory across^[32,47], vegetation sensitivity^[36], and water memory length and strength^[39].

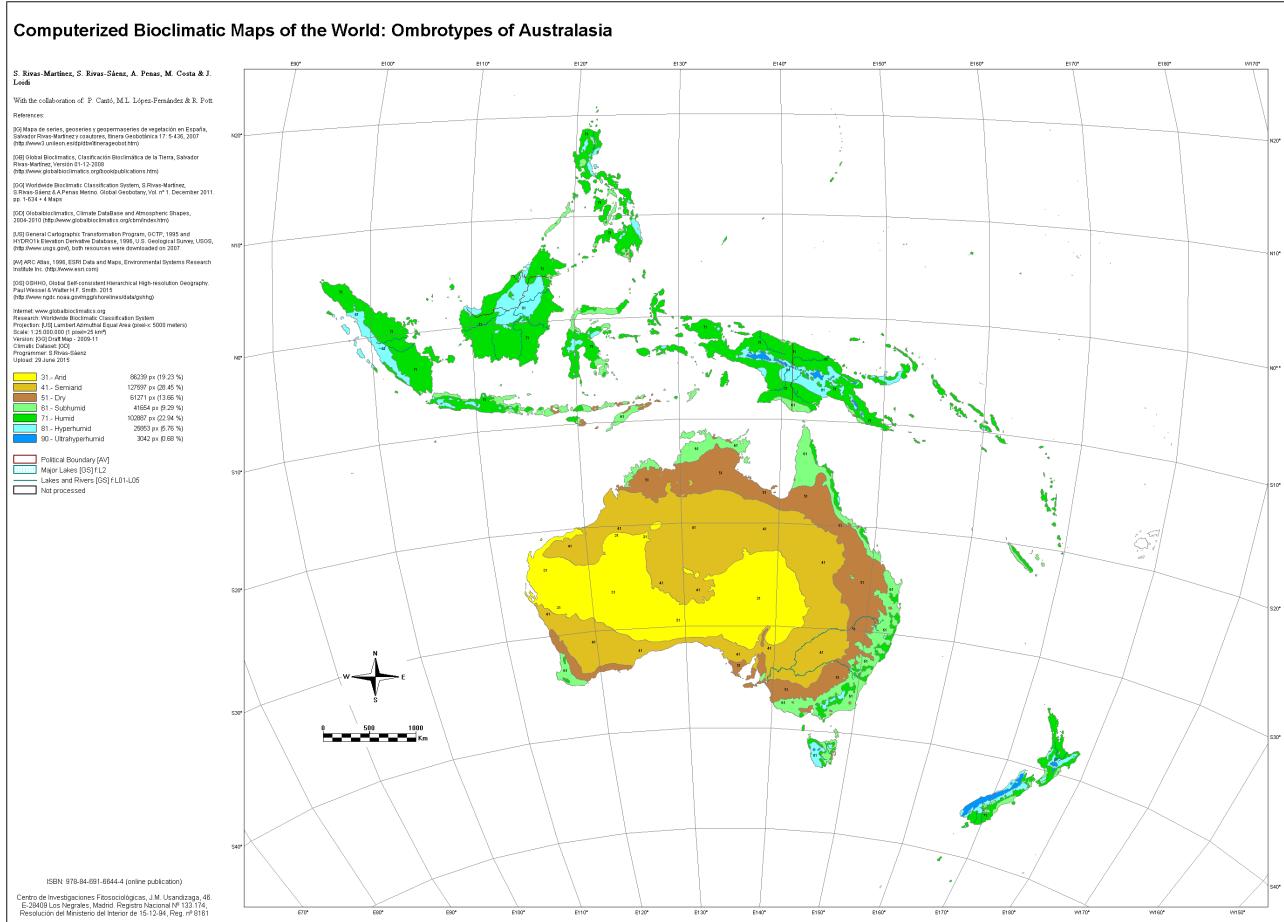


Figure 2.11: Australasian Ombrotypes - Regions of mean precipitation rates of biological relevance across Australasia according to Rivas-Martínez et al. [127] and retrieved via the Worldwide Bioclimatic Classification System [131].

See figures A.8 and A.9 for an overview of GIMMS NDVI, ERA5 data, and TRY PFT data (both raw and extrapolated) across Australia.

2.3 Analyses

All analyses, with the exception of python, bash, and matlab scripts to obtain ERA5 data, have been carried out in R. All codes needed to reproduce the analysis at the core of this thesis have been included to this document in section A.3. Note that the computation can be sped up by altering the `Cores` argument to enable parallel processing via the `foreach` and `doparallel`^[132] packages.

2.3.1 Identifying Vegetation Memory

2.3.1.1 Data Preparation

Monthly Maximum Composites

After retrieving GIMMS NDVI 3g data using the `gimms` package in R, bi-weekly NDVI data is aggregated to monthly composites using the *maximum composite* method (see Chunk 8). Doing so is a method for removing data uncertainty due to atmospheric interference which has seen application in recent remote-sensing-based studies of vegetation memory and resilience^[11,49,111].

Downscaling

In order to calculate vegetation memory effects, vegetation data and abiotic data should be represented at the same spatial and temporal resolution. Whilst both NDVI and ERA5 data are available at monthly intervals (after pre-processing as already described), their spatial resolutions differ (e.g. $\sim 9.27\text{km} \times \sim 9.27\text{km}$, vs. $30\text{km} \times 30\text{km}$, respectively; see tables 2.1 and 2.2).

There are two ways of remedying this spatial mismatch: (1) aggregating GIMMS data to coarser resolution of ERA5 data, or (2) downscaling of ERA5 data to GIMMS resolution. Whilst aggregating data to coarser resolutions is much easier and less computationally expensive, I have opted to apply **downscaling** to the ERA5 data used within my study so as to retain valuable information within the GIMMS NDVI 3g data set. This will allow more precise vegetation memory identification and result in more biologically relevant patterns of vegetation memory characteristics.

A large host of downscaling methods is available and before choosing any method one has to settle on whether to perform *statistical* or *dynamic* downscaling^[105,133]:

1. *Statistical Downscaling* is centred around statistical links between global and local climate patterns^[134], comparatively computationally cheap and easily transferable to different study regions but suffers from a strong dependency on the choice of predictors/co-variates^[135].
2. *Dynamic Downscaling* is built around regional climate models being implemented into global climate models^[134], produces results based on physically consistent processes but computationally expensive^[135].

This study is focussed on three study regions (i.e. the Iberian Region, the Caatinga, and Australia) and five ERA5 variables (Tair, Qsoil1, Qsoil2, Qsoil3, and Qsoil4) in monthly intervals from 1981 to 2015. ERA5 data from 1981 is being downscaled to match GIMMS resolution to calculate cumulative lags of memory effect drivers (see section 2.3.1.2). With this, downscaling needs to be performed three times (study regions), for five variables, each with 420 individual time steps (twelve months per year times 35 years). In total, the downscaling process is thus required to be run 6,300 times. Using a computationally expensive method belonging to the group of dynamical downscaling approaches is thus undesirable and I have elected to employ a **statistical downscaling** method to match ERA5 data with GIMMS resolution.

More specifically, I am using **Kriging** - a method that is well-understood and has long been used in non-biological sciences for geostatistical interpolation purposes^[136]. Kriging is a two-step process. First, one establishes statistical

relationships between data which is to be kriged at its native resolution with covariate data at the same resolution. The second step sees the extrapolation of these relationships using target resolution covariate data. The way in which Kriging improves over other statistical downscaling methods centred around these two steps lies in the fact that the Kriging methodology not only *extrapolates* relationships but *residuals* as well (see figure 2.12 for a visual representation).

Therefore, Kriging within my analyses requires three inputs: (1) ERA5 data at native resolution, (2) HWSD covariates (see table A.2) at ERA5 resolution, and (3) HWSD covariates (see table A.2) at GIMMS resolution to produce sets of ERA5 data at GIMMS resolution. Kriging operations are built around formulae which establish response and predictor relationships. My kriging approaches are specified as follows:

$$Var_{ERA5} = \alpha + \sum_{i=1}^{14} (\beta_i * Cov_{HWSD;i}) \quad (2.3)$$

with Var_{ERA5} identifying any of the five ERA5 variables Tair, Qsoil1, Qsoil2, Qsoil3, or Qsoil4 at any given monthly interval between January 1981 and December 2015. $Cov_{HWSD;i}$ indexes the i^{th} HWSD covariate ranging from elevation and slope aspects to slope incline levels. Due to computational expense the kriging procedure only considers interaction effects between slope incline levels and slope aspects (see Chunk 9).

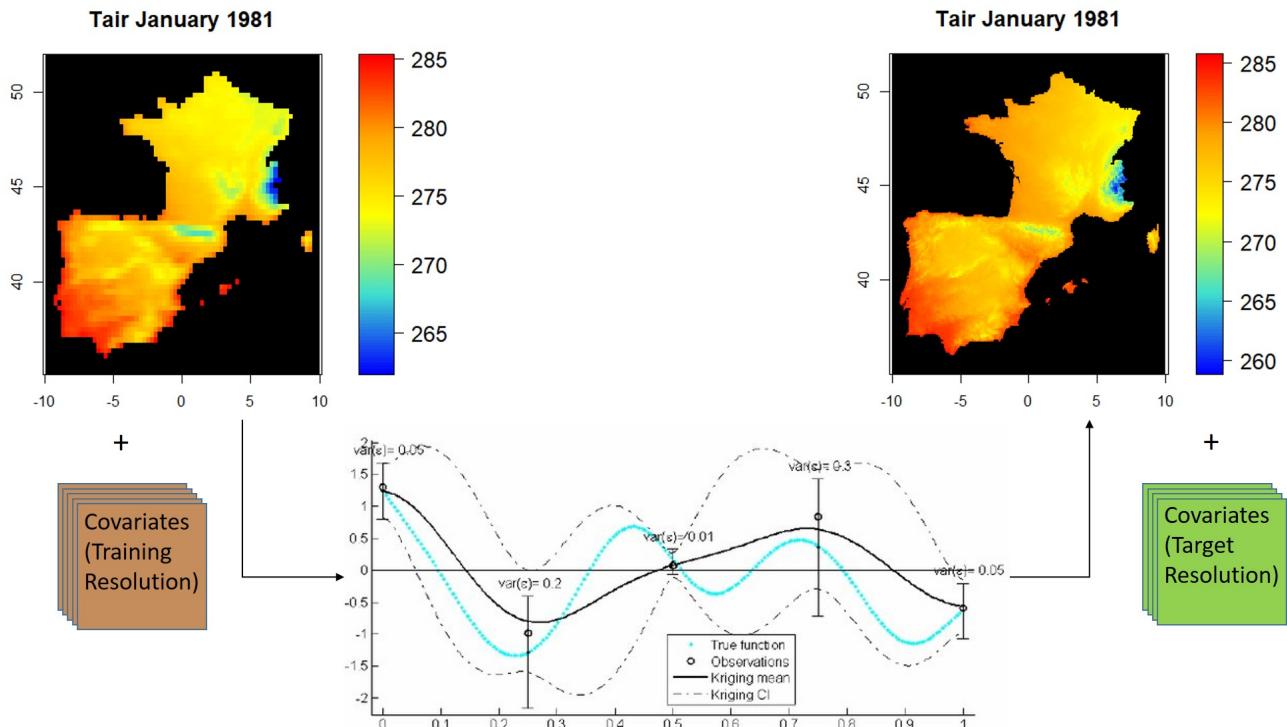


Figure 2.12: Kriging Concept - Statistical downscaling effects of Tair for the time step of January 1981 using HWSD covariate data across the Iberian Region. Diagram source: Le Riche et al., 2012^[137]. Some parts of this figure have been generated via Chunk 27.

For an in-depth mathematical explanation of the Kriging methodology, see Hengl, 2011^[136]. A practical example of its use can be retrieved in Lichtenstern, 2013^[138].

2.3.1.2 Memory Models

Models of vegetation memory within this study come in two flavours:

1. *Vegetation Memory Coefficients* are being computed to identify intrinsic and extrinsic forcing factors.
2. *Model Comparisons* and *Variance Partitioning* are employed to assess relative importance of different factors.

Vegetation Memory Coefficients

The analyses used to identify vegetation memory characteristics in this study have been informed heavily by DeKeersmaecker et al.'s work^[32]. In general, the vegetation memory models of this study are built upon the following basic specification:

$$NDVI_t = \beta_{t-1} * NDVI_{[t-1]} + \beta_{Qsoil} * Qsoil_{k;m} + \beta_{Tair} * Tair_t \quad (2.4)$$

with $NDVI_t$ being standardised NDVI anomalies at time step t ^[32], $NDVI_{[t-1]}$ identifying the NDVI anomalies at time step $t - 1$, $Qsoil_{k;m}$ denoting Qsoil data at depth level k (translating to Qsoil1- Qsoil4) and cumulative lag of standardised anomalies of lag m , and $Tair_t$ denoting Tair data at time t . See table 2.4 for an interpretation of the model coefficients. The duality of memory effects as *intrinsic* and *extrinsic* in nature as proposed by Ogle et al^[41] is embraced by my models as follows:

1. **Intrinsic Memory** is identified as coefficients of $NDVI_{[t-1]}$ ^[11,32] (see model formula 2.4).
2. **Extrinsic Memory** are implemented via ERA5 data. See section 2.1.2.1 for the rationale behind the inclusion of the different variables and model formula 2.4 for how they are included in vegetation memory models.
 - (a) $Tair$ is implemented as an *instantaneous* effect on plant performance.
 - (b) $Qsoil1$ - $Qsoil4$ effects are implemented as *cumulative lag* effects ranging from instantaneous impacts to lags on annual time windows^[49].

Table 2.4: Interpretation of Memory Model Coefficients - Biological Interpretation of Vegetation Memory Coefficients as portrayed in model formula 2.4.

Coefficient	Magnitude	Sign
β_{t-1}	Absolute values depict the velocity at which systems return to equilibrium/pre-disturbance state. Large absolute values indicate low resilience (i.e. slow return).	<i>Positive</i> - NDVI anomalies resemble previous ones. NDVI anomalies gradually diminish over time. <i>Negative</i> - NDVI anomalies resemble previous ones, but with the opposite sign. The return to pre-disturbance is characterised through oscillations.
β_{Qsoil}	Absolute values depict the resistance to anomalies in Qsoil. Large absolute values indicate low resistance (i.e. strong vegetation responses) to Qsoil anomalies.	<i>Positive</i> - Wetter soil conditions than average induce positive NDVI anomalies; drier soil conditions than average induce negative NDVI anomalies. <i>Negative</i> - Drier soil conditions than average induce positive NDVI anomalies; wetter soil conditions than average induce negative NDVI anomalies.
β_{Tair}	Absolute values depict the resistance to anomalies in air temperature. Large absolute values indicate low resistance (i.e. strong vegetation responses) to air temperature anomalies.	<i>Positive</i> - Warmer air temperature than average induces positive NDVI anomalies; colder air temperature than average induces negative NDVI anomalies. <i>Negative</i> - Colder air temperature than average induces positive NDVI anomalies; warmer air temperature than average induces negative NDVI anomalies.

The coefficients of the above drivers of vegetation memory (see table 2.4 and formula 2.4) are identified for each data pixel in the data rasters of the three study regions in four separate models (one for each Qsoil layer).

Doing so entails handling an immense amount of data which defies individual data preparation and model inspection and so one has to make generate a generalised, and automated approach. A visual representation of the automated modelling approach used within this study can be seen in figure 2.13 and is carried out for each pixel as follows using the code contained within Chunk 10:

1. Data for each variable (NDVI, Tair, Qsoil) is **extracted and detrended** to avoid effects of changing abiotic conditions over long time-series^[32] using linear detrending with the pracma package in R.
2. Detrended data is **standardised to Z-Scores** to obtain deviations of monthly means/monthly anomalies for each variable^[32]:

$$\text{Anomaly}_i = \frac{\text{Detrended}_i - \overline{\text{Detrended}_{month}}}{SD_{\text{Detrended},month}} \quad (2.5)$$

with i indexing individual, detrended data records. In the case of NDVI, one additionally calculates monthly means of untreated NDVI data across the entire data range. See figure 2.15 for an examplatory overview of NDVI data treatment.

3. Calculation of lagged effects:

- (a) $NDVI_{t-1}$ is calculated from Z-Score NDVI data.
- (b) Cumulative lags of Qsoil data are established for lags ranging from 0 (instantaneous effects) to annual effects (aggregated over twelve months of Qsoil Z-scores) in steps on one month at a time. See figure ?? for an examplatory overview of Qsoil data treatment.
- (c) Tair data is implemented as instantaneous effects and so no lagged effects have to be calculated.
4. Variables in nature are often collinear^[63,139]. Failing to address this issue results in masking of information which might influence our understanding of processes in nature significantly^[140]. One method of circumventing this issue lies with **PCA**. More specifically, if regression modelling is the target, one may wish to employ **PCA regression** as a three-step process^[140] for each of the cumulative Qsoil lags across all four Qsoil layers effectively adding the necessity for a model selection step:
 - (a) Z-Score data for NDVI, Tair, and Qsoil are fed to PCA via the vegan package in R. See figure 2.16 for an examplatory overview of data and a PCA result.
 - (b) Regression is performed as follows:

$$NDVI_t = \beta_1 * PC_1 + \beta_2 * PC_2 + \beta_3 * PC_3 \quad (2.6)$$

with $NDVI_t$ representing NDVI anomalies which have been set to NA (skipped in models) in months for which $Thresholds_i < 0.1$ with $Thresholds_i = \overline{Raw_{NDVI,month}}$ ^[11]. PC_1 through PC_3 and β_1 through β_3 indicate principal components 1 through 3 and coefficients of their effects in the model respectively.

- (c) Model selection is performed to identify the cumulative soil lag which presents the most explanatory power through comparison of model Akaike Information Criterion (AIC) values. The model with the AIC closest to 0 is said to be performing the best^[141]. This results in a proxy of Qsoil memory length of local vegetation.
- (d) The regression coefficients β_1 through β_3 can be back-transformed to represent PCA input variable effects (see formula 2.4) using PCA loadings and PCA model coefficients (see figure 2.17).

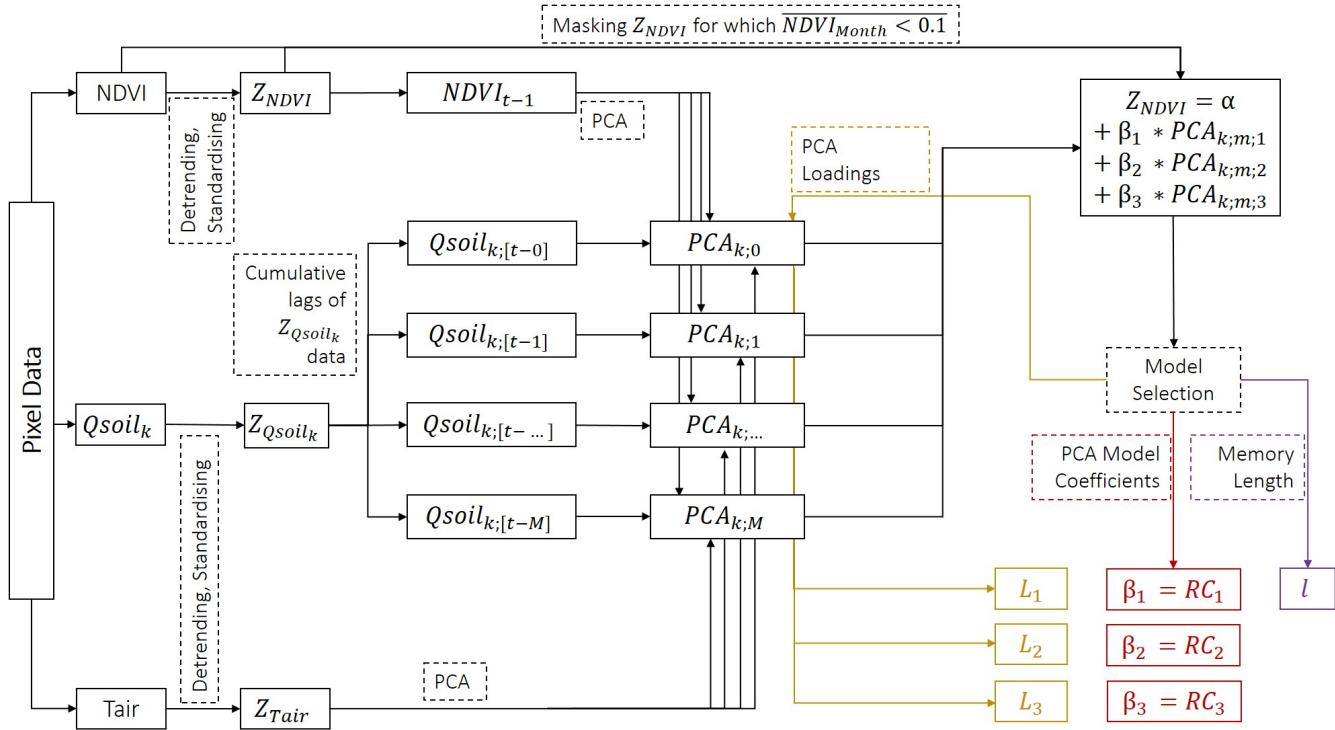


Figure 2.13: Vegetation Memory Model Flowchart - Visual workflow of pixel-wise iterated vegetation memory model. m denotes the currently considered cumulative lag of Qsoil data with M being the maximum cumulative lag. Qsoil layers Qsoil₁- Qsoil₄ are identified via k . L_1 through L_3 are the loadings of each detrended and standardised variable (NDVI, Tair, and Qsoil) onto the principal components 1 through 3, respectively. PCA model coefficients are identified as β_1 through β_3 . l denotes the cumulative Qsoil lag offering the most explanatory power in terms of AIC values of PCA regression models and is thus a proxy for vegetation memory in terms of Qsoil.

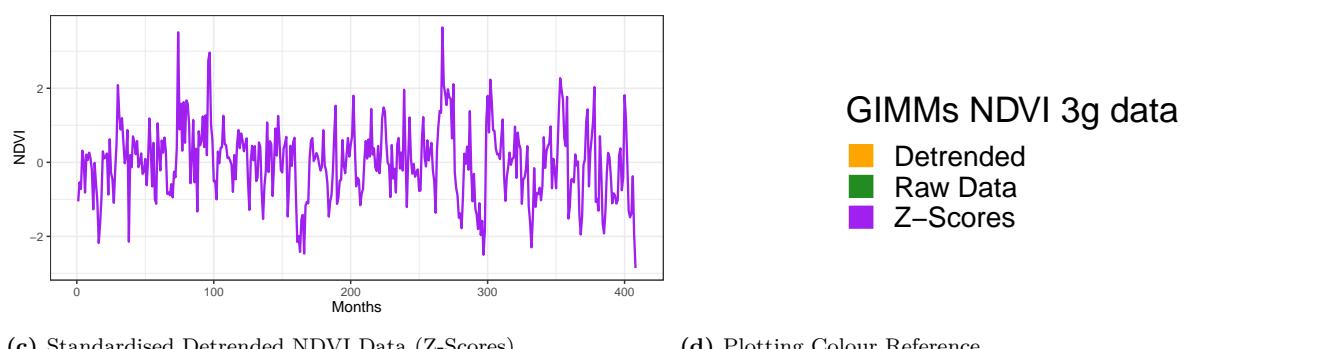
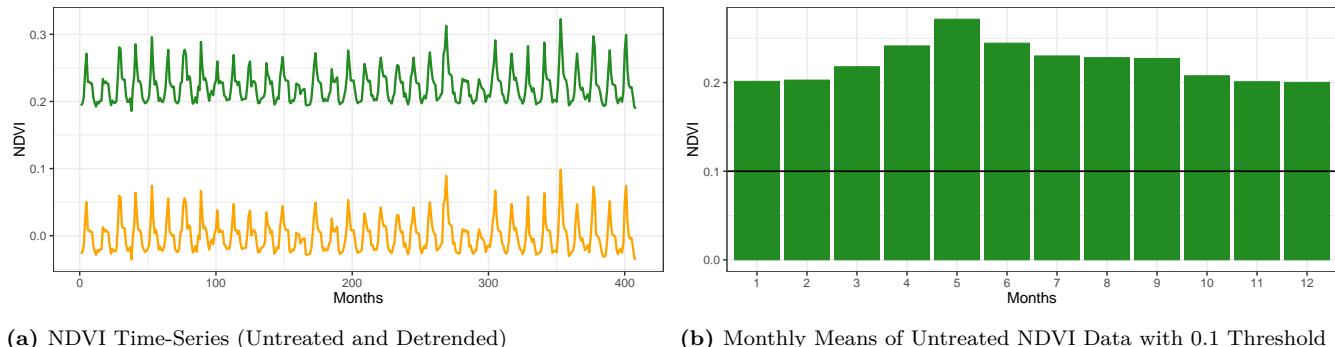


Figure 2.14: Memory Model NDVI Data Treatment - Overview of NDVI data treatment as outlined in figure 2.13 and explained above. The data presented here represents a single pixel in the Iberian dryland region (see figure 2.16). Figure established via Chunk 25.

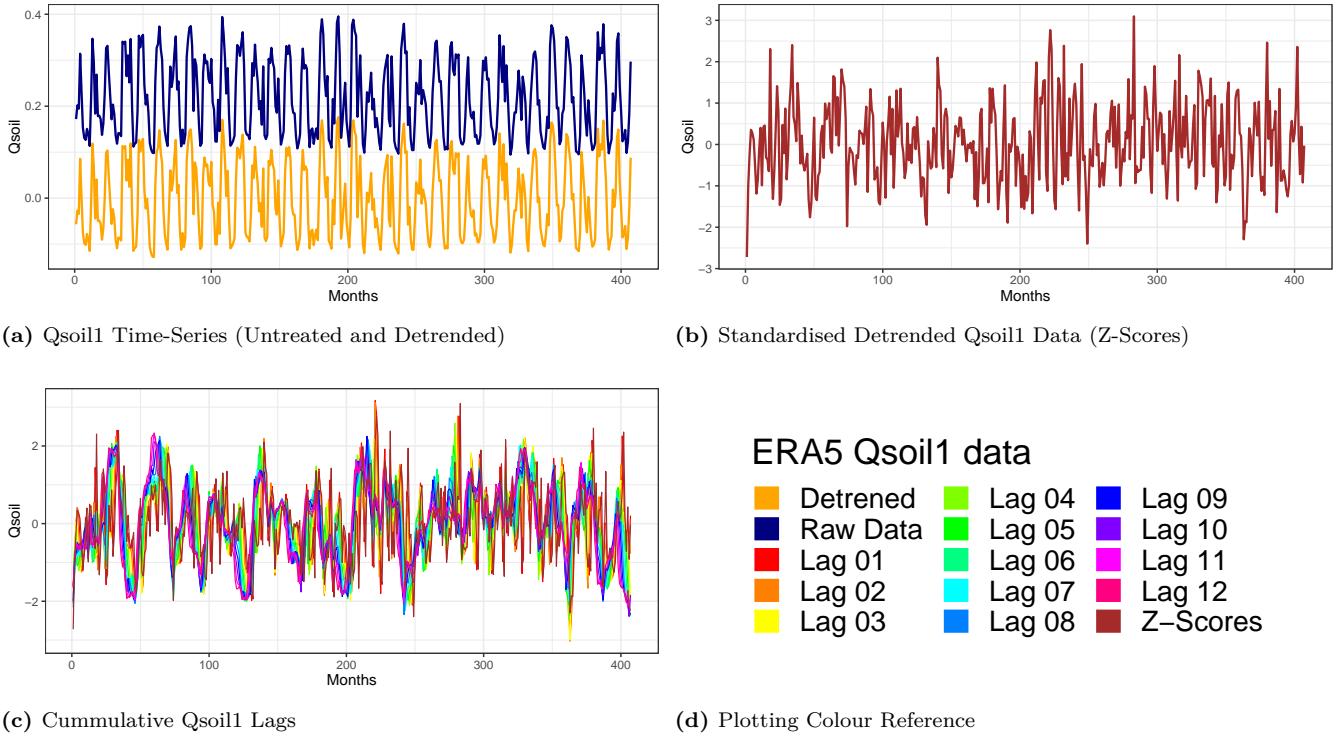


Figure 2.15: Memory Model Qsoil Data Treatment - Overview of Qsoil data treatment as outlined in figure 2.13 and explained above. Only Qsoil data is represented. The data presented here represents a single pixel in the Iberian dryland region (see figure 2.16). Figure established via Chunk 26.

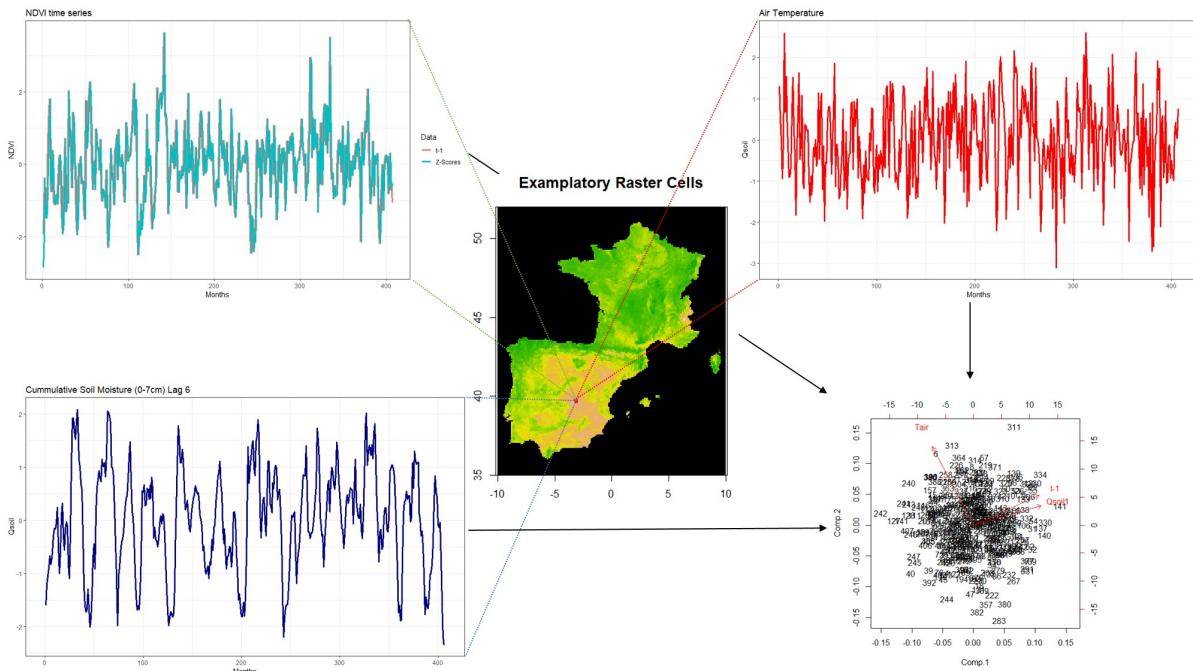


Figure 2.16: Exemplatory Vegetation Memory PCA - Overview of Z-Score data for $NDVI_{t-1}$, $Qsoil_{1;6}$, and $Tair$ for a single pixel in the dryland region of the Iberian study region (red colouring on NDVI background in central plot) as well as their representation in a PCA. Also pictured: $NDVI_t$ Z-Scores required for PCA regression models (see figure 2.13 and 2.17). Some parts of this figure have been generated via Chunk 27.

Model Formula: $NDVI_t = PC_1 + PC_2 + PC_3$

- $NDVI_t$ NDVI anomaly at month t
- PC_1 First principal component
- PC_2 Second principal component
- PC_3 Third principal component

	PC1	PC2	PC3
$NDVI_{t-1}$	2.5	-1.14	1.95
$Qsoil_{1,6}$	2.6	-0.73	-2.03
$Tair$	-1.6	-3.00	-0.25
Model Coefficients	1.77	-0.50	0.71

$$C_p = \sum_{i=1}^3 (L_{p;i} * RC_i)$$

- C_p Coefficient of variable p
- i Principal component counter
- $L_{p;i}$ Loading of variable p on principal component i
- RC_i Model coefficient for principal component i

	PC1	PC2	PC3	Coefficient
$NDVI_{t-1}$	4.41	0.57	1.39	6.4
$Qsoil_{1,6}$	4.57	0.37	-1.45	4.7
$Tair$	-2.79	1.50	-0.18	-1.5

Σ

Figure 2.17: PCA Regression Coefficients - Theoretical back-calculation of regression coefficients from PCA regression coefficients as lined out by Zuur et. al^[140]. The data presented here (PCA loadings, PCA model coefficients, and final variable coefficients) represents a single pixel in the Iberian dryland region (see figure 2.16).

By following the vegetation memory modelling procedure outlined in figures 2.13 and 2.17 and explained above for each study region one obtains rasters containing the following information for each pixel:

1. **Vegetation Memory Coefficients** as established in formula 2.4.
2. **Vegetation Memory Length** in regards to Qsoil1, Qsoil2, Qsoil3, and Qsoil4.

Model Comparisons

Assessments of differences in relative importance of vegetation memory coefficients in individual models and between models serve to answer the following questions:

1. **Which model variable exerts the greatest influence on vegetation anomalies?** - To answer this question, I am comparing absolute values of variable coefficients across all pixels *within* each *model* individually, separately for each study region.
2. **Which Qsoil layer is the most biologically influential?** - The answer to this question can be retrieved by comparing absolute values of Qsoil coefficients for all pixel per model *between* all four *models* for each study region.

Differences in absolute variable coefficient values within and between models has been assessed using Mann-Whitney-U Test (`wilcox.test(..., paired = FALSE)` in R). I have chosen the Mann-Whitney U-Test, since vegetation memory coefficient data cannot be expected to be normal distributed, nor to fall onto symmetrical distributions (hence, one should contrast median values rather than mean values). The code for these assessments can be retrieved in Chunk 10 (`CoeffScaling`).

Although allowing for region-wide generalisations, these assessments of statistical significance can not be used to identify or display spatial patterns of relative memory coefficient importance.

Variance Partitioning

Variance partitioning is a model-driven method of assessing relative importance of vegetation memory model variables. As opposed to Mann-Whitney U model comparisons, variance partitioning can be carried out for each pixel in my study region data rasters individually. This, in turn, results in the identification of patterns of relative model predictor importance.

It was my goal to assess the relative importance of intrinsic and extrinsic vegetation memory components. I have identified Qsoil layers to be of special interest in representing extrinsic vegetation memory (see section 3). Therefore, I am assessing the relative information contained in the vegetation memory model predictors $NDVI_{t-1}$ (intrinsic memory) and $Qsoil_{k;m}$ (extrinsic soil moisture memory of layer k and cumulative lag m).

Zuur et al.^[140] present a method developed by Legendre & Legendre^[142] for the purpose of variance partitioning between two explanatory variables. This approach (represented in figure 2.18) is carried out as follows:

1. Apply the full regression model $Z_{NDVI} = NDVI_{t-1} + Qsoil_{k;m}$ and obtain R^2 (the coefficient of determination). This is R_{Full}^2 and equal to all explained variance. Unexplained variance can then be calculated as $1 - R_{Full}^2$
2. Obtain R^2 of the model $Z_{NDVI} = NDVI_{t-1}$. This is $R_{NDVI_{t-1}}^2$ and equal to all variance explained by $NDVI_{t-1}$.
3. Obtain R^2 of the model $Z_{NDVI} = Qsoil_{k;m}$. This is $R_{Qsoil_{k;m}}^2$ and equal to all variance explained by $Qsoil_{k;m}$.

The variance shared between $NDVI_{t-1}$ and $Qsoil_{k;m}$ - R_{Shared}^2 can then be calculated as:

$$R_{Shared}^2 = R_{NDVI_{t-1}}^2 + R_{Qsoil_{k;m}}^2 - R_{Full}^2 \quad (2.7)$$

The pure information contained within $NDVI_{t-1}$ (R_{t-1}^2) and $Qsoil_{k;m}$ ($R_{k;m}^2$) can then be calculated as follows:

$$R_{t-1}^2 = R_{NDVI_{t-1}}^2 - R_{Shared}^2 \quad (2.8)$$

$$R_{k;m}^2 = R_{Qsoil_{k;m}}^2 - R_{Shared}^2 \quad (2.9)$$

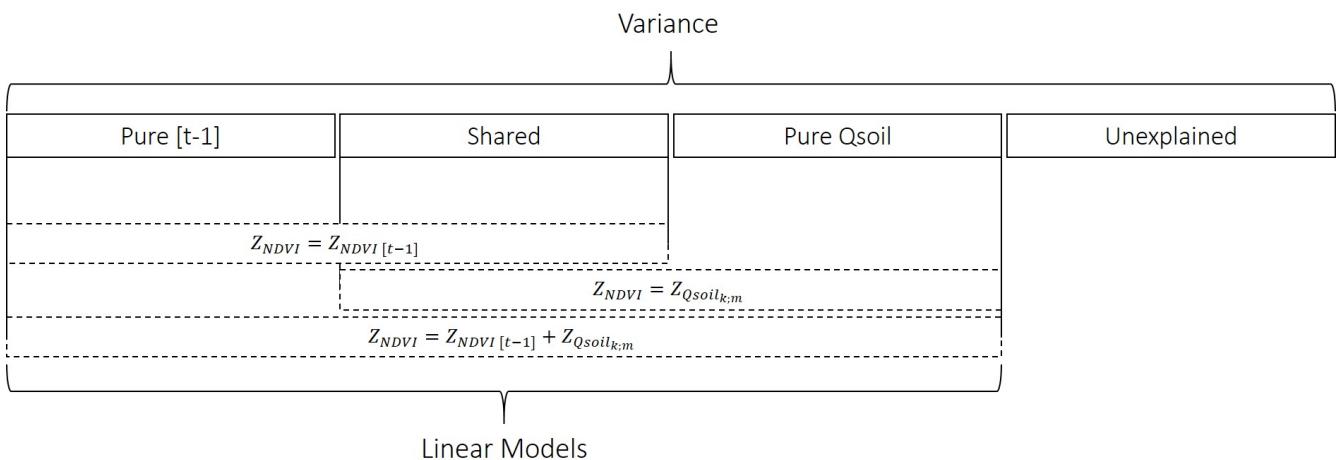


Figure 2.18: Variance Partitioning - Partitioning of variance in pure influence of $NDVI_{t-1}$ (Pure $[t - 1]$), $Qsoil_{k;m}$ (Pure $Qsoil$, with k identifying the Qsoil layer, and m denoting the cumulative lag of Qsoil data), shared variance between the two (Shared), and residuals (Unexplained). The figure concept has been lifted from Zuur et al.^[140] and adjusted to reflect the purpose of this study. Model specifications are contained within dashed boxes.

This form of variance partitioning is carried out for each pixel across all study regions and contained in the vegetation memory model output of Chunk 10 alongside PCA regression models.

2.3.2 Linking Vegetation Memory and Plant Function

My study employs PFT and LHT data to assess possible links between vegetation memory and expressions of plant function (*research goal 2*).

2.3.2.1 Plant Functional Traits

PFT data has been obtained via TRY on 07/08/18 in geo-referenced data table format.

Data Extraction

Vegetation memory characteristics are stored as raster data sets to retain spatial patterns of memory coefficients. Therefore, PFT data needs rasterising for comparability. PFT data is extracted and rasterised in two ways:

1. **Raw geo-referenced PFT records.** Geo-referenced TRY data points are aggregated to rasters of GIMMS resolution using a mean function (if multiple PFT records fall onto the same raster cell, the mean value of these is assigned to the cell). This results in rasters of low data-coverage but peer-reviewed data records.
2. **Species-specific mean PFT records.** These are extracted for each species within the TRY data set to enable PFT mapping as depicted in figure 1.3.

Data extraction of PFT records is handled via Chunk 11.

Data Mapping

Data mapping is carried out via Chunk 11 and extrapolates species-specific PFT mean values according to GBIF occurrence data (see figure 1.3). Using the approach presented by Ordóñez & Svenning^[60], I am establishing rasters of mean PFT expressions by:

1. Assigning species-specific mean PFT values to all cells of individual species occurrence according to GBIF.
2. Computing the mean PFT value for each cell in the rasters.
3. Removing all cells whose values exceed the upper 95% quantile of mean PFT records in these final mean rasters to remove outliers which may be due to sampling bias.

This results in rasters of high data coverage but at a loss of data reliability.

Models

I use regression models to assess causal links between PFT expressions (both raw and mean-extrapolated).

2.3.2.2 Life History Traits

LHT data has been obtained via COMPADRE on 28/01/19 in geo-referenced table format.

Data Extraction

Target COMPADRE LHTs are extracted using Chunk 12 by aggregating geo-referenced LHT records to rasters of GIMMS resolution using a mean function for each study region.

Models

Like with PFTs, I use regression models to assess causal links between LHTs expressions (both raw and mean-extrapolated).

3. Results

3.1 Iberian Region

3.1.1 Vegetation Memory

3.1.1.1 Memory Models

Soil Moisture (0-7cm)

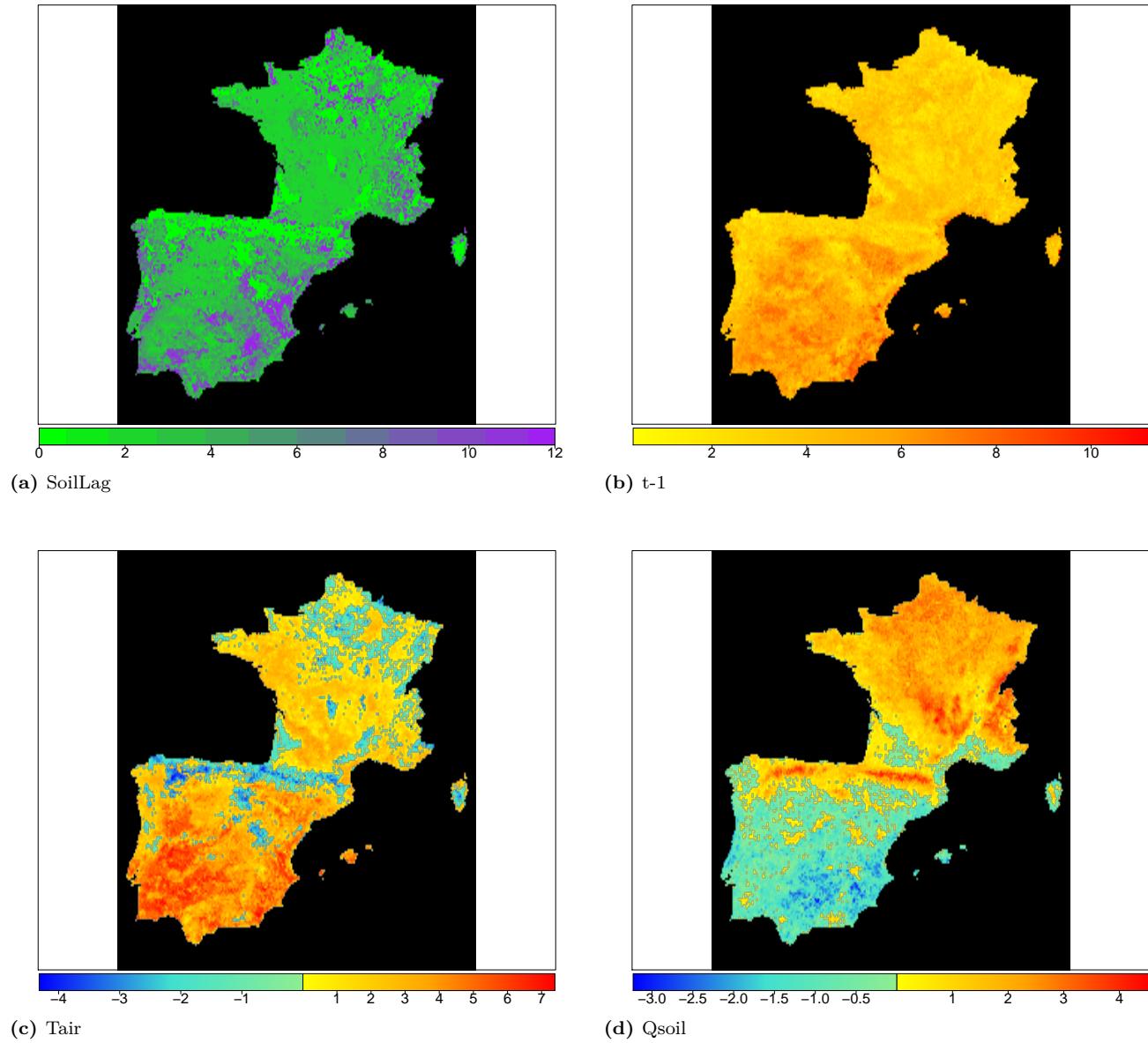
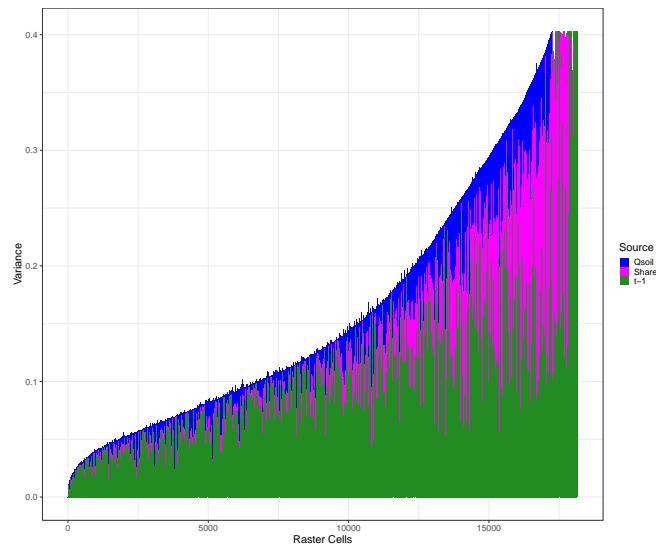
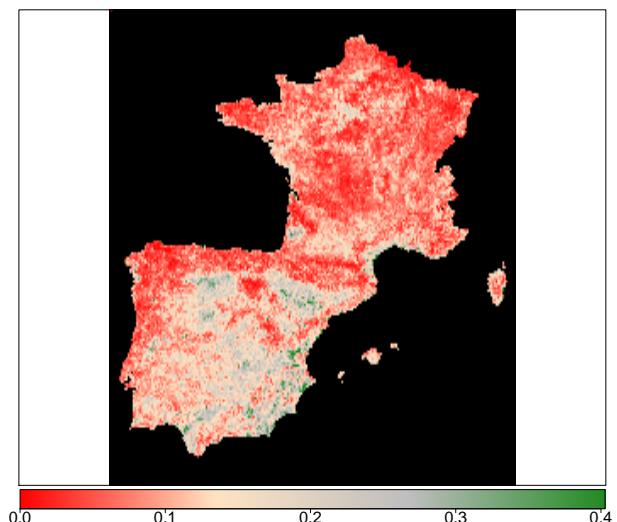


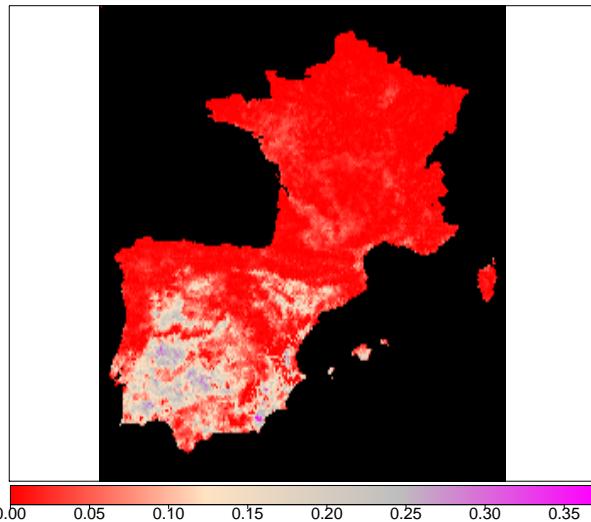
Figure 3.1: Short - Long.



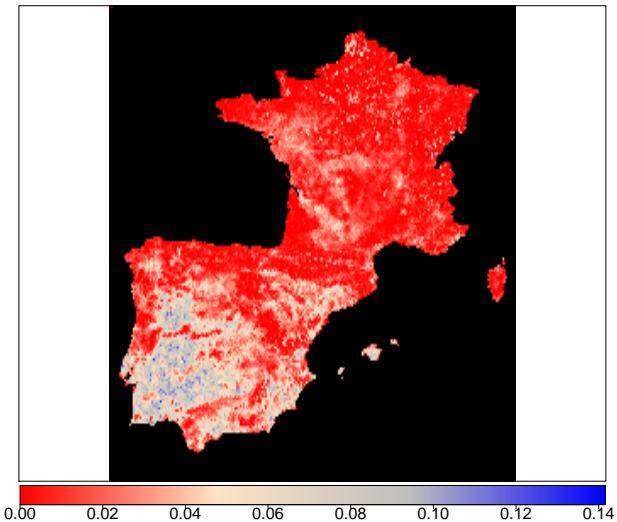
(a) Total



(b) t-1



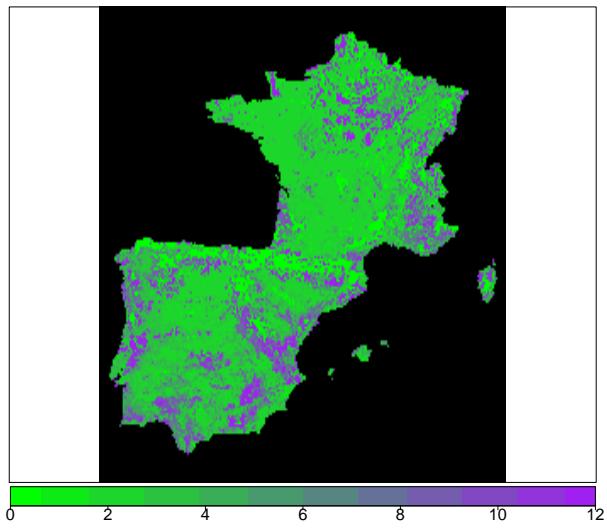
(c) Shared



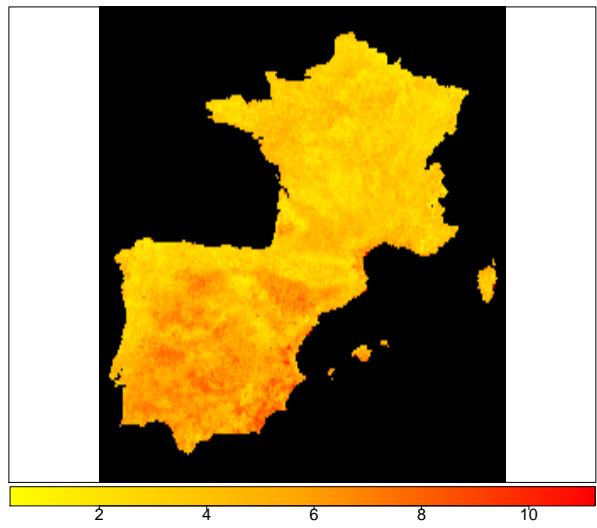
(d) Qsoil

Figure 3.2: Short - Long.

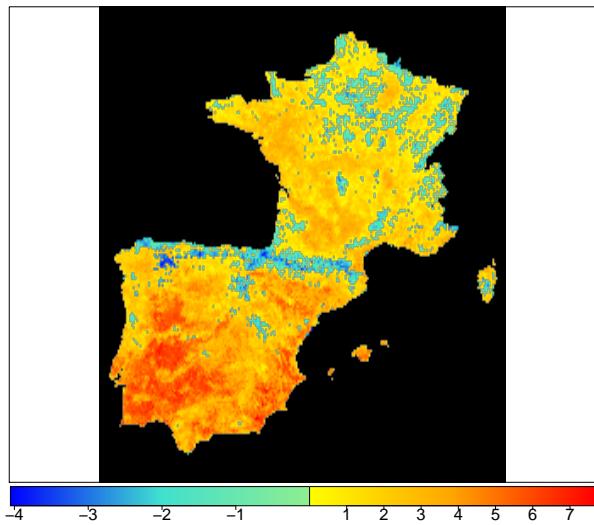
Utest

Soil Moisture (7-28cm)

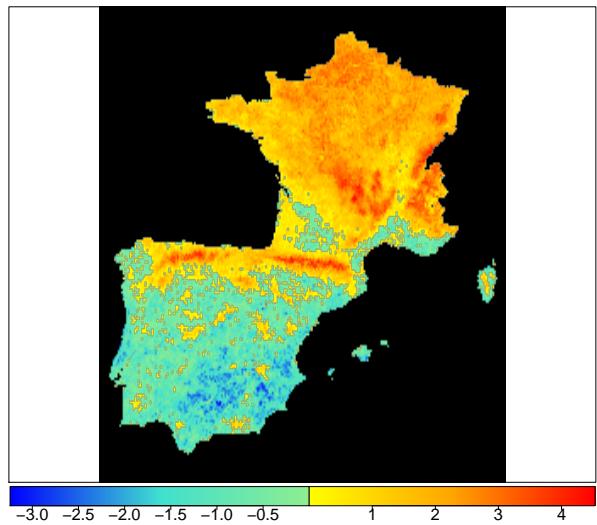
(a) SoilLag



(b) t-1

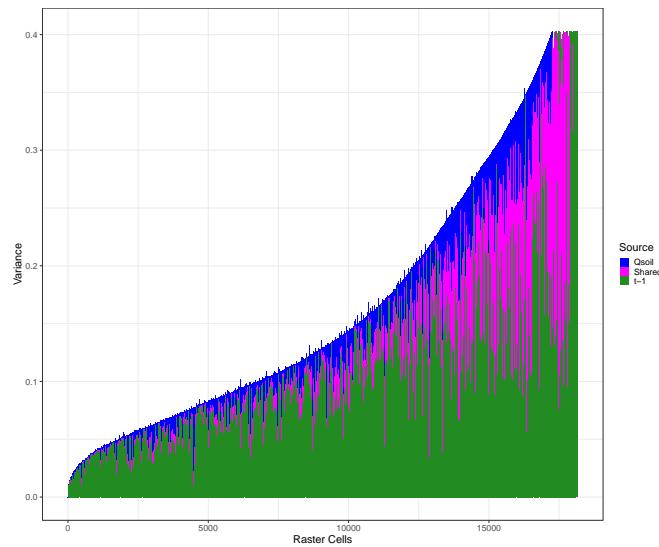


(c) Tair

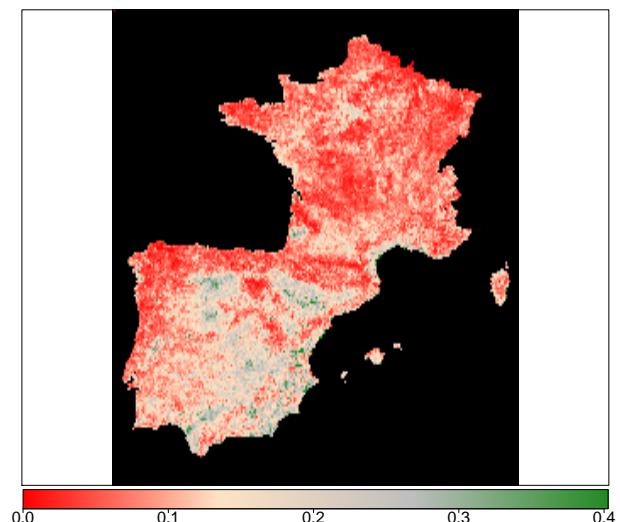


(d) Qsoil

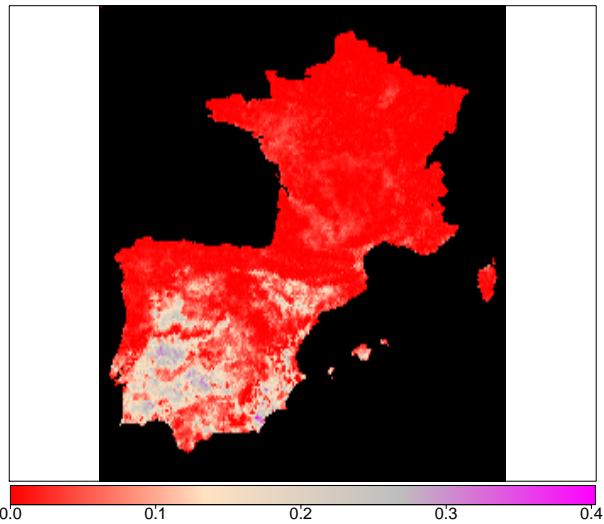
Figure 3.3: Short - Long.



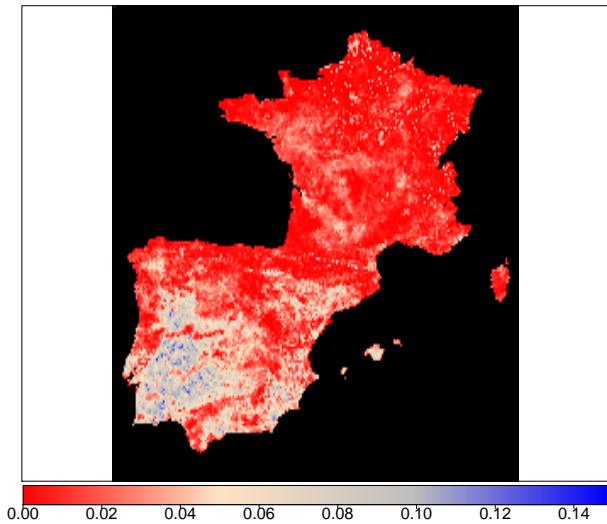
(a) Total



(b) t-1



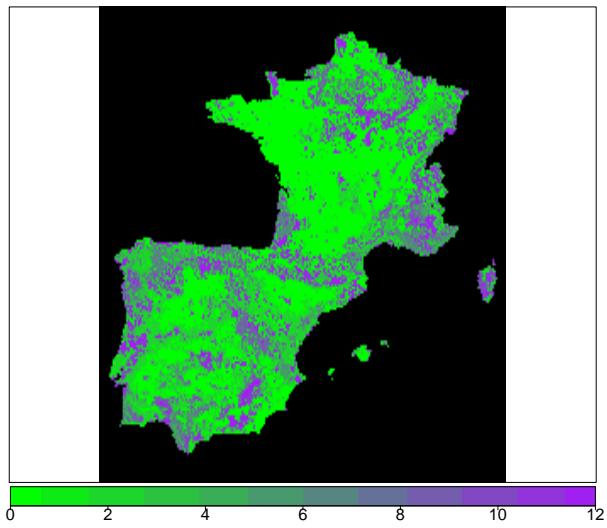
(c) Shared



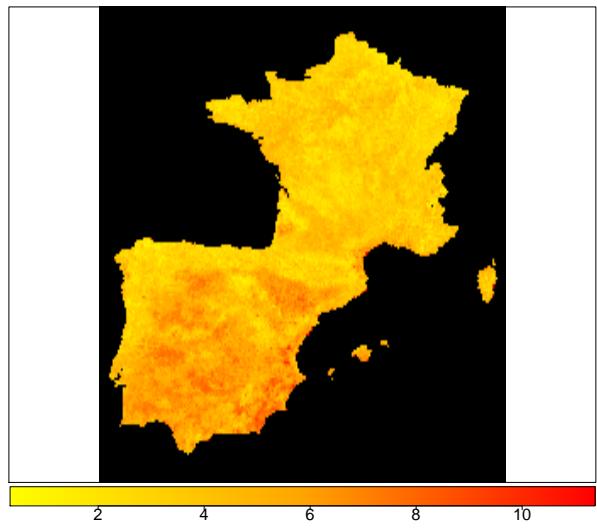
(d) Qsoil

Figure 3.4: Short - Long.

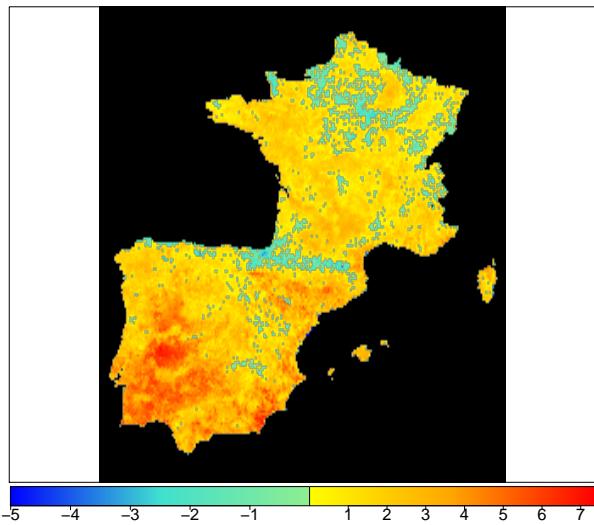
Utest

Soil Moisture (28-100cm)

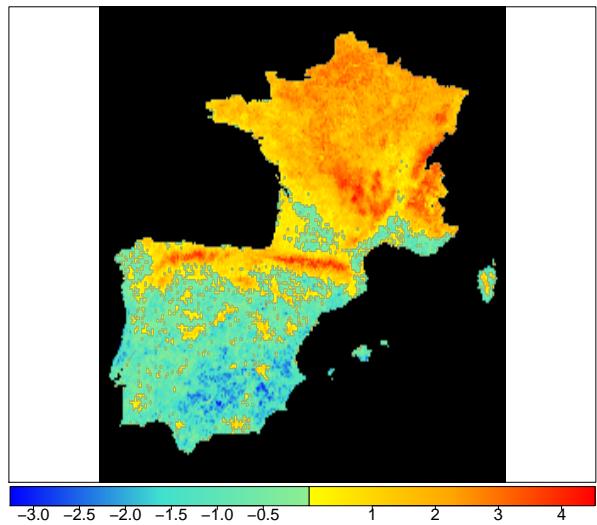
(a) SoilLag



(b) t-1



(c) Tair



(d) Qsoil

Figure 3.5: Short - Long.

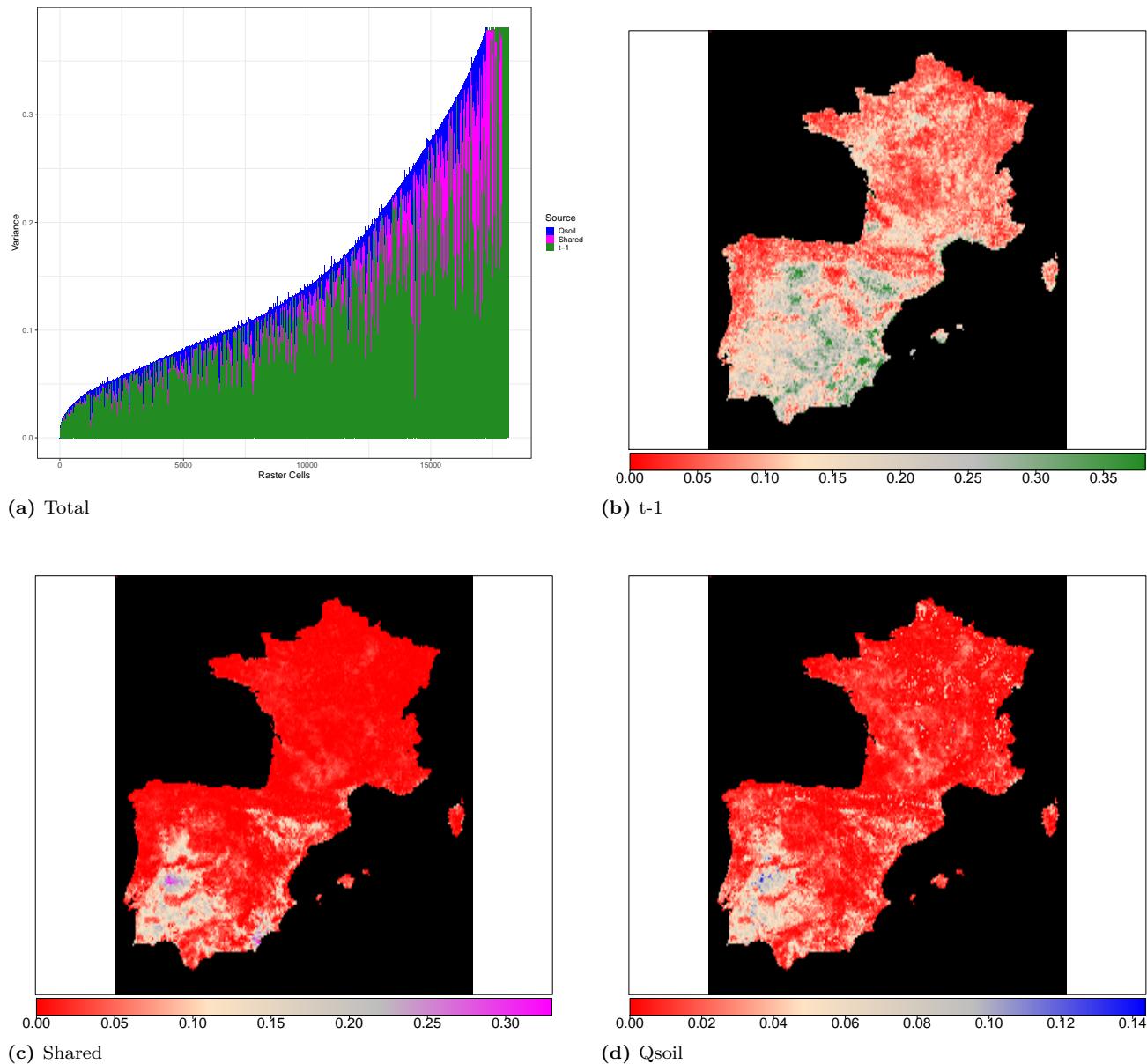
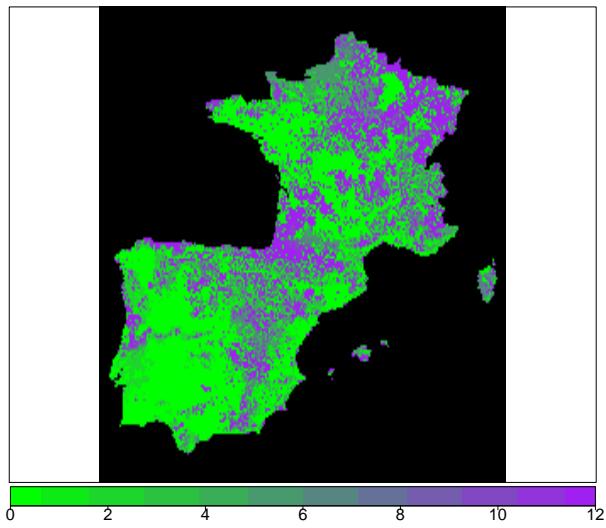
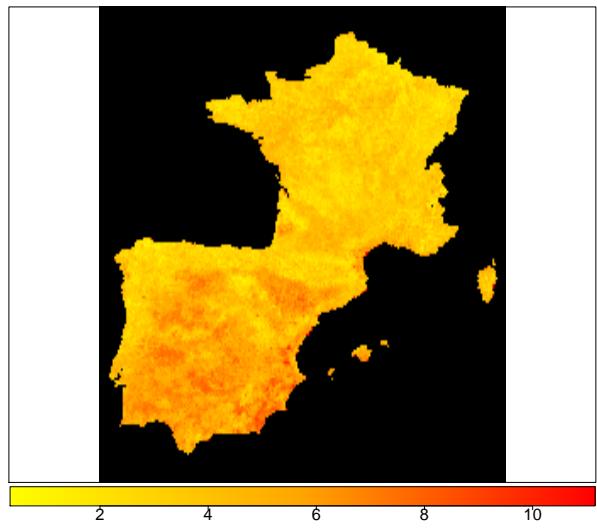


Figure 3.6: Short - Long.

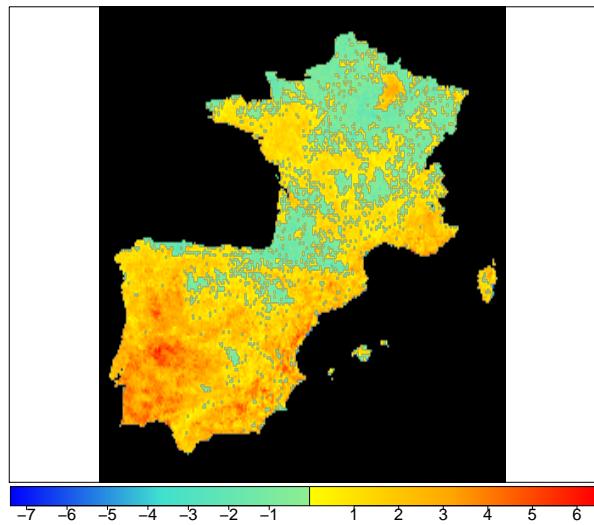
Utest

Soil Moisture (100-255cm)

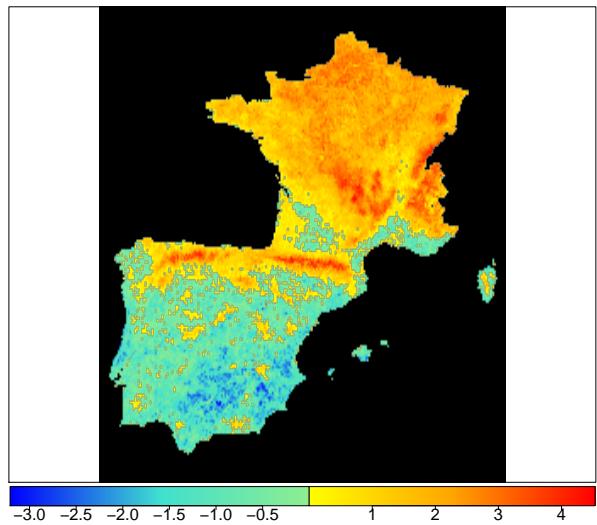
(a) SoilLag



(b) t-1

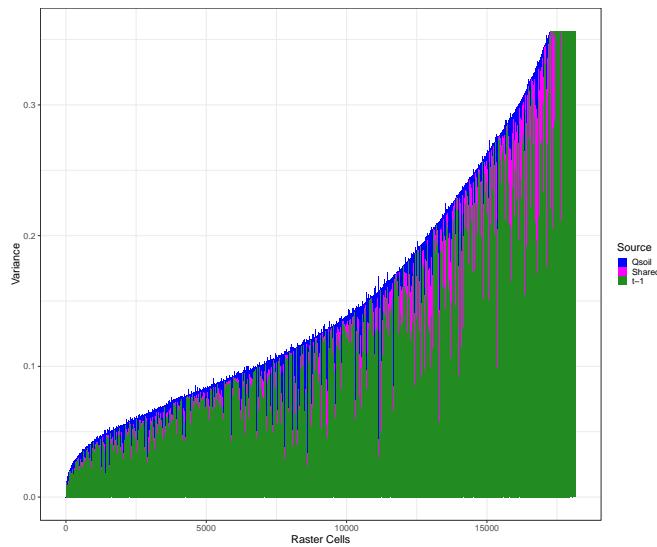


(c) Tair

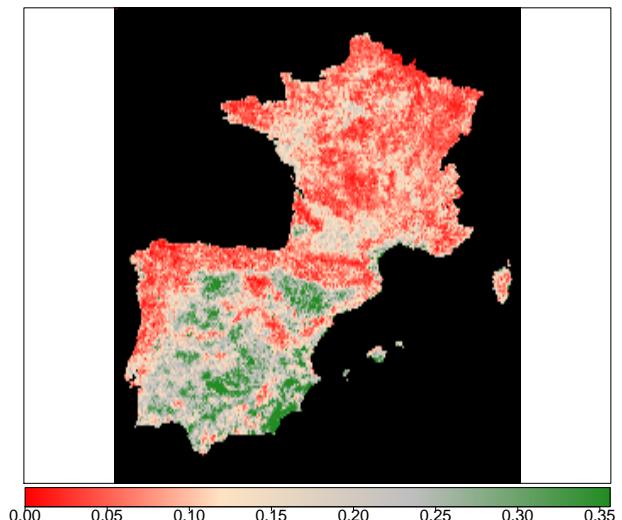


(d) Qsoil

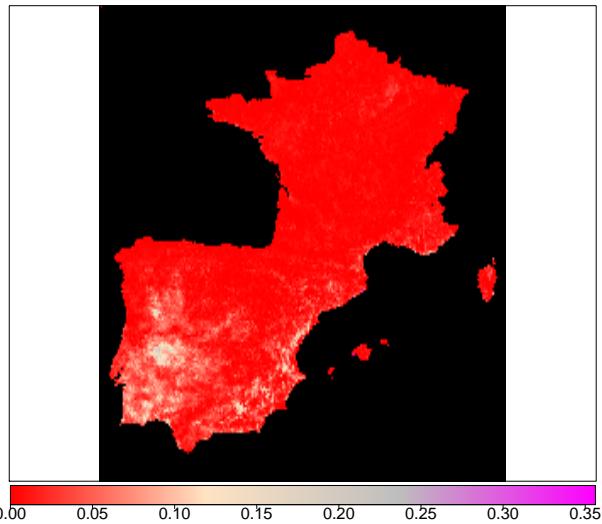
Figure 3.7: Short - Long.



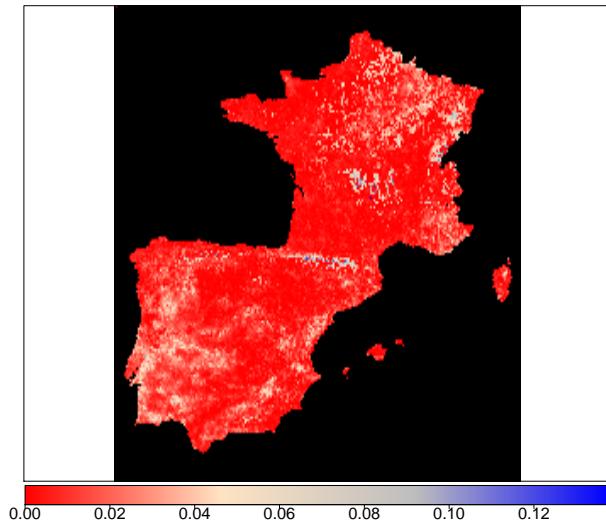
(a) Total



(b) t-1



(c) Shared



(d) Qsoil

Figure 3.8: Short - Long.

Utest

3.1.1.2 Memory Coefficients

Soil Layer Comparison (Plot scaled + Uttest)

3.1.1.3 Vegetation Memory Sensitivity

Coefficients vs mean of qsoil 1-4, tair, and ndvi + lags vs means?

3.1.2 Plant Function

3.1.2.1 Plant Functional Traits

3.1.2.2 Life History Traits

3.2 Caatinga, Brazil

3.2.1 Vegetation Memory

3.2.2 Plant Function

3.3 Australia

3.3.1 Vegetation Memory

3.3.2 Plant Function

3.4 Global Effects

3.4.1 Vegetation Memory

3.4.2 Plant Function

4. Discussion

4.1 Iberian Region

4.1.1 Vegetation Memory

4.1.2 Plant Function

4.2 Caatinga, Brazil

4.2.1 Vegetation Memory

4.2.2 Plant Function

4.3 Australia

4.3.1 Vegetation Memory

4.3.2 Plant Function

4.4 Global Effects

4.4.1 Vegetation Memory

4.4.2 Plant Function

5. Conclusion

Outlook

1. Soil moisture-driven drought indices^[143]?
2. Use BIEN data

Bibliography

- [1] Johnstone, J. F. *et al.* Changing disturbance regimes, ecological memory, and forest resilience. *Frontiers in Ecology and the Environment* **14**, 369–378 (2016). arXiv:1011.1669v3.
- [2] Senf, C. & Seidl, R. Natural disturbances are spatially diverse but temporally synchronized across temperate forest landscapes in Europe. *Global Change Biology* **24**, 1201–1211 (2018). 0608246v3.
- [3] Scheffer, M. Foreseeing tipping points. *Nature* **467**, 411–412 (2010).
- [4] Xu, C., Van Nes, E. H., Holmgren, M., Kéfi, S. & Scheffer, M. Local Facilitation May Cause Tipping Points on a Landscape Level Preceded by Early-Warning Indicators. *The American Naturalist* **186**, E81–E90 (2015).
- [5] Vasilakopoulos, P. & Marshall, C. T. Resilience and tipping points of an exploited fish population over six decades. *Global Change Biology* **21**, 1834–1847 (2015).
- [6] Côté, I. M. & Darling, E. S. Rethinking Ecosystem Resilience in the Face of Climate Change. *PLoS Biology* **8**, e1000438 (2010).
- [7] Scheffer, M., Carpenter, S., Foley, J. a., Folke, C. & Walker, B. Catastrophic shifts in ecosystems. *Nature* **413**, 591–596 (2001).
- [8] Genkai-Kato, M. Regime shifts: catastrophic responses of ecosystems to human impacts. *Ecological Research* **22**, 214–219 (2007).
- [9] CBD. Aichi Biodiversity Targets (2013).
- [10] Holling, C. S. Resilience and Stability of Ecological Systems. *Annual Review of Ecology and Systematics* **4**, 1–23 (1973).
- [11] Liu, L., Zhang, Y., Wu, S., Li, S. & Qin, D. Water memory effects and their impacts on global vegetation productivity and resilience. *Scientific Reports* **8**, 1–9 (2018).
- [12] Mumby, P. J. & Anthony, K. R. N. Resilience metrics to inform ecosystem management under global change with application to coral reefs. *Methods in Ecology and Evolution* 1–9 (2015).
- [13] Seidl, R., Spies, T. A., Peterson, D. L., Stephens, S. L. & Hicke, J. A. REVIEW: Searching for resilience: addressing the impacts of changing disturbance regimes on forest ecosystem services. *Journal of Applied Ecology* **53**, 120–129 (2016).
- [14] Scheffer, M., Carpenter, S. R., Dakos, V. & van Nes, E. H. Generic Indicators of Ecological Resilience: Inferring the Chance of a Critical Transition. *Annual Review of Ecology, Evolution, and Systematics* **46**, 145–167 (2015).
- [15] Standish, R. J. *et al.* Resilience in ecology: Abstraction, distraction, or where the action is? *Biological Conservation* **177**, 43–51 (2014).
- [16] Gunderson, L. H. Ecological Resilience—In Theory and Application. *Annual Review of Ecology and Systematics* **31**, 425–439 (2000).
- [17] Newton, A. C. Biodiversity Risks of Adopting Resilience as a Policy Goal. *Conservation Letters* **9**, 369–376 (2016).
- [18] Carpenter, S., Walker, B., Andries, J. M. & Abel, N. From Metaphor to Measurement: Resilience of What to What? *Ecosystems* **4**, 765–781 (2001).

- [19] Brown, E. D. & Williams, B. K. Resilience and Resource Management. *Environmental Management* **56**, 1416–1427 (2015).
- [20] Scheffer, M. & Carpenter, S. R. Catastrophic regime shifts in ecosystems: linking theory to observation. *Trends in Ecology & Evolution* **18**, 648–656 (2003).
- [21] Rist, L. *et al.* Applying resilience thinking to production ecosystems. *Ecosphere* **5**, art73 (2014).
- [22] Tabacchi, E., Steiger, J., Corenblit, D., Monaghan, M. T. & Planty-Tabacchi, A.-M. Implications of biological and physical diversity for resilience and resistance patterns within Highly Dynamic River Systems. *Aquatic Sciences* **71**, 279–289 (2009).
- [23] Mumby, P. J., Chollett, I., Bozec, Y. M. & Wolff, N. H. Ecological resilience, robustness and vulnerability: How do these concepts benefit ecosystem management? (2014). URL <http://dx.doi.org/10.1016/j.cosust.2013.11.021>.
- [24] Angeler, D. G. & Allen, C. R. Quantifying resilience. *Journal of Applied Ecology* **53**, 617–624 (2016).
- [25] Foley, J. A., Coe, M. T., Scheffer, M. & Wang, G. Regime Shifts in the Sahara and Sahel: Interactions between Ecological and Climatic Systems in Northern Africa. *Ecosystems* **6**, 524–532 (2003).
- [26] Mouillot, D., Graham, N. A. J., Villéger, S., Mason, N. W. H. & Bellwood, D. R. A functional approach reveals community responses to disturbances. *Trends in Ecology and Evolution* **28**, 167–177 (2013). 2072.
- [27] Grimm, V. & Wissel, C. Babel, or the ecological stability discussions: an inventory and analysis of terminology and a guide for avoiding confusion. *Oecologia* **109**, 323–334 (1997).
- [28] Wardle, D. A. & Jonsson, M. Long-term resilience of above- and belowground ecosystem components among contrasting ecosystems. *Ecology* **95**, 1836–1849 (2014).
- [29] Seddon, A. W. R., Froyd, C. A., Witkowski, A. & Willis, K. J. A quantitative framework for analysis of regime shifts in a Galápagos coastal lagoon. *Ecology* **95**, 3046–3055 (2014).
- [30] Mori, A. S., Furukawa, T. & Sasaki, T. Response diversity determines the resilience of ecosystems to environmental change. *Biological Reviews* **88**, 349–364 (2013).
- [31] Selkoe, K. A. *et al.* Principles for managing marine ecosystems prone to tipping points. *Ecosystem Health and Sustainability* **1**, 1–18 (2015).
- [32] De Keersmaecker, W. *et al.* A model quantifying global vegetation resistance and resilience to short-term climate anomalies and their relationship with vegetation cover. *Global Ecology and Biogeography* **24**, 539–548 (2015).
- [33] Hodgson, D., McDonald, J. L. & Hosken, D. J. What do you mean, ‘resilient’? *Trends in Ecology & Evolution* **30**, 503–506 (2015).
- [34] Peterson, G., Allen, C. R. & Holling, C. S. Original Articles: Ecological Resilience, Biodiversity, and Scale. *Ecosystems* **1**, 6–18 (1998). 9610022.
- [35] Allen, C. R. *et al.* Quantifying spatial resilience. *Journal of Applied Ecology* **53**, 625–635 (2016).
- [36] Seddon, A. W. R., Macias-Fauria, M., Long, P. R., Benz, D. & Willis, K. J. Sensitivity of global terrestrial ecosystems to climate variability. *Nature* **531**, 229–232 (2016).
- [37] Redmond, M. D., Weisberg, P. J., Cobb, N. S. & Clifford, M. J. Woodland resilience to regional drought: Dominant controls on tree regeneration following overstorey mortality. *Journal of Ecology* **106**, 625–639 (2018).
- [38] Papagiannopoulou, C. *et al.* A non-linear Granger-causality framework to investigate climate–vegetation dynamics. *Geoscientific Model Development* **10**, 1945–1960 (2017).
- [39] Gazol, A., Camarero, J. J., Anderegg, W. R. L. & Vicente-Serrano, S. M. Impacts of droughts on the growth resilience of Northern Hemisphere forests. *Global Ecology and Biogeography* **26**, 166–176 (2017).

- [40] Walter, J., Jentsch, A., Beierkuhnlein, C. & Kreyling, J. Ecological stress memory and cross stress tolerance in plants in the face of climate extremes. *Environmental and Experimental Botany* **94**, 3–8 (2013).
- [41] Ogle, K. *et al.* Quantifying ecological memory in plant and ecosystem processes. *Ecology Letters* **18**, 221–235 (2015).
- [42] Munné-Bosch, S. & Alegre, L. Cross-stress tolerance and stress "memory" in plants: An integrated view. *Environmental and Experimental Botany* **94**, 1–2 (2013). arXiv:1011.1669v3.
- [43] Papagiannopoulou, C. *et al.* Vegetation anomalies caused by antecedent precipitation in most of the world. *Environmental Research Letters* **12**, 074016 (2017).
- [44] Rudgers, J. A. *et al.* Climate sensitivity functions and net primary production: A framework for incorporating climate mean and variability. *Ecology* **99**, 576–582 (2018). 0608246v3.
- [45] Esper, J., Schneider, L., Smerdon, J. E., Schöne, B. R. & Büntgen, U. Signals and memory in tree-ring width and density data. *Dendrochronologia* **35**, 62–70 (2015).
- [46] Nyström, M. & Folke, C. Spatial Resilience of Coral Reefs. *Ecosystems* **4**, 406–417 (2001).
- [47] De Keersmaecker, W. *et al.* Assessment of Regional Vegetation Response to Climate Anomalies: A Case Study for Australia Using GIMMS NDVI Time Series between 1982 and 2006. *Remote Sensing* **9**, 34 (2017).
- [48] Cole, L. E. S., Bhagwat, S. A. & Willis, K. J. Recovery and resilience of tropical forests after disturbance. *Nature Communications* **5**, 3906 (2014).
- [49] Vicente-Serrano, S. M. *et al.* Response of vegetation to drought time-scales across global land biomes. *Proceedings of the National Academy of Sciences* **110**, 52–57 (2013).
- [50] Westoby, M. & Wright, I. J. Land-plant ecology on the basis of functional traits. *Trends in Ecology & Evolution* **21**, 261–268 (2006). arXiv:1011.1669v3.
- [51] De Boeck, H. J. *et al.* Patterns and drivers of biodiversity-stability relationships under climate extremes. *Journal of Ecology* **106**, 890–902 (2018).
- [52] Truchy, A., Angeler, D. G., Sponseller, R. A., Johnson, R. K. & McKie, B. G. Linking Biodiversity, Ecosystem Functioning and Services, and Ecological Resilience. In *Advances in Ecological Research*, vol. 53, 55–96 (Elsevier Ltd., 2015), 1 edn.
- [53] Mace, G. M. *et al.* Approaches to defining a planetary boundary for biodiversity. *Global Environmental Change* **28**, 289–297 (2014).
- [54] Lavorel, S. & Garnier, E. Predicting changes in community composition and ecosystem functioning from plant traits: revisiting the Holy Grail. *Functional Ecology* **16**, 545–556 (2002).
- [55] Nock, C. A., Vogt, R. J. & Beisner, B. E. Functional Traits. In *eLS*, 1–8 (John Wiley & Sons, Ltd, Chichester, UK, 2016).
- [56] Laughlin, D. C. The intrinsic dimensionality of plant traits and its relevance to community assembly. *Journal of Ecology* **102**, 186–193 (2014).
- [57] Ackerly, D. D. Community Assembly, Niche Conservatism, and Adaptive Evolution in Changing Environments. *International Journal of Plant Sciences* **164**, S165–S184 (2003).
- [58] Wright, I. J. *et al.* Modulation of leaf economic traits and trait relationships by climate. *Global Ecology and Biogeography* **14**, 411–421 (2005).
- [59] Bonser, S. P. Form defining function: interpreting leaf functional variability in integrated plant phenotypes. *Oikos* **114**, 187–190 (2006).
- [60] Ordonez, A. & Svenning, J.-C. Geographic patterns in functional diversity deficits are linked to glacial-interglacial climate stability and accessibility. *Global Ecology and Biogeography* **24**, 826–837 (2015).

- [61] Díaz, S. *et al.* The global spectrum of plant form and function. *Nature* **529**, 167–171 (2016).
- [62] Yachi, S. & Loreau, M. Biodiversity and ecosystem productivity in a fluctuating environment: The insurance hypothesis. *Proceedings of the National Academy of Sciences* **96**, 1463–1468 (1999).
- [63] Butler, E. E. *et al.* Mapping local and global variability in plant trait distributions. *Proceedings of the National Academy of Sciences* **114**, E10937–E10946 (2017).
- [64] Violle, C. *et al.* The return of the variance: intraspecific variability in community ecology. *Trends in Ecology & Evolution* **27**, 244–252 (2012).
- [65] Dantas, V. d. L., Pausas, J. G., Batalha, M. A., de Paula Loiola, P. & Cianciaruso, M. V. The role of fire in structuring trait variability in Neotropical savannas. *Oecologia* **171**, 487–494 (2013).
- [66] Kattge, J. *et al.* TRY - a global database of plant traits. *Global Change Biology* **17**, 2905–2935 (2011).
- [67] Pérez-Harguindeguy, N. *et al.* New handbook for standardised measurement of plant functional traits worldwide. *Australian Journal of Botany* **61**, 167 (2013).
- [68] Gerhard, B. & Kattge, J. Try plant trait data base (2019). URL <https://www.try-db.org/TryWeb/About.php>.
- [69] Enquist, B. J., Condit, R., Peet, R. K., Schildhauer, M. & Thiers, B. The Botanical Information and Ecology Network (BIEN): Cyberinfrastructure for an integrated botanical information network to investigate the ecological impacts of global climate change on plant biodiversity. Tech. Rep. (2009).
- [70] Botanical information and ecology network (2019). URL <http://bien.nceas.ucsb.edu/bien/biendata/>.
- [71] Salguero-Gómez, R. *et al.* The comadre Plant Matrix Database: an open online repository for plant demography. *Journal of Ecology* **103**, 202–218 (2015).
- [72] Salguero-Gómez, R. *et al.* Fast–slow continuum and reproductive strategies structure plant life-history variation worldwide. *Proceedings of the National Academy of Sciences* **113**, 230–235 (2016).
- [73] Comadre plant matrix data base (2019). URL <https://www.comadre-db.org/Data/Comadre>.
- [74] Bruelheide, H. *et al.* Global trait–environment relationships of plant communities. *Nature Ecology & Evolution* **2**, 1906–1917 (2018).
- [75] Gbif - global biodiversity information facility (2019). URL <https://www.gbif.org/>.
- [76] Sandel, B. *et al.* Estimating themissing species bias in plant trait measurements. *Journal of Vegetation Science* **26**, 828–838 (2015).
- [77] Newbold, T., Butchart, S. H. M., Şekercioğlu, Ç. H., Purves, D. W. & Scharlemann, J. P. W. Mapping Functional Traits: Comparing Abundance and Presence-Absence Estimates at Large Spatial Scales. *PLoS ONE* **7**, e44019 (2012).
- [78] Prestele, R. *et al.* Hotspots of uncertainty in land-use and land-cover change projections: a global-scale model comparison. *Global Change Biology* **22**, 3967–3983 (2016).
- [79] Peterson, G. D. Contagious Disturbance, Ecological Memory, and the Emergence of Landscape Pattern. *Ecosystems* **5**, 329–338 (2002).
- [80] Bell, D. M., Cohen, W. B., Reilly, M. & Yang, Z. Visual interpretation and time series modeling of Landsat imagery highlight drought’s role in forest canopy declines. *Ecosphere* **9**, e02195 (2018).
- [81] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria (2016). URL <https://www.R-project.org/>.
- [82] Nioti, F., Xystrakis, F., Koutsias, N. & Dimopoulos, P. A Remote Sensing and GIS Approach to Study the Long-Term Vegetation Recovery of a Fire-Affected Pine Forest in Southern Greece. *Remote Sensing* **7**, 7712–7731 (2015).
- [83] Prince, S. D. & Goward, S. N. Global Primary Production: A Remote Sensing Approach. *Journal of*

- Biogeography* **22**, 815 (1995).
- [84] Garrouette, E., Hansen, A. & Lawrence, R. Using NDVI and EVI to Map Spatiotemporal Variation in the Biomass and Quality of Forage for Migratory Elk in the Greater Yellowstone Ecosystem. *Remote Sensing* **8**, 404 (2016).
- [85] Cammarano, D., Fitzgerald, G., Casa, R. & Basso, B. Assessing the Robustness of Vegetation Indices to Estimate Wheat N in Mediterranean Environments. *Remote Sensing* **6**, 2827–2844 (2014).
- [86] Friedl, M. *et al.* Global land cover mapping from MODIS: algorithms and early results. *Remote Sensing of Environment* **83**, 287–302 (2002).
- [87] Evrendilek, F. & Gulbeyaz, O. Deriving Vegetation Dynamics of Natural Terrestrial Ecosystems from MODIS NDVI/EVI Data over Turkey. *Sensors* **8**, 5270–5302 (2008).
- [88] Colditz, R. R. *et al.* Generation and analysis of the 2005 land cover map for Mexico using 250m MODIS data. *Remote Sensing of Environment* **123**, 541–552 (2012).
- [89] Krishna Prasad, V., Badarinath, K. V. S. & Eaturu, A. Spatial patterns of vegetation phenology metrics and related climatic controls of eight contrasting forest types in India – analysis from remote sensing datasets. *Theoretical and Applied Climatology* **89**, 95–107 (2007).
- [90] Stow, D. A. *et al.* Remote sensing of vegetation and land-cover change in Arctic Tundra Ecosystems. *Remote Sensing of Environment* **89**, 281–308 (2004).
- [91] Chen, B., Chen, J. M. & Ju, W. Remote sensing-based ecosystem–atmosphere simulation scheme (EASS)—Model formulation and test with multiple-year data. *Ecological Modelling* **209**, 277–300 (2007).
- [92] Lieng, E. *et al.* Classification of Land Cover Using Decision Trees and Multiple Reference Data Sources. Tech. Rep. (2005).
- [93] Friedl, M. A. *et al.* MODIS Collection 5 global land cover: Algorithm refinements and characterization of new datasets. *Remote Sensing of Environment* **114**, 168–182 (2010).
- [94] Devadas, R., Lamb, D. W., Simpfendorfer, S. & Backhouse, D. Evaluating ten spectral vegetation indices for identifying rust infection in individual wheat leaves. *Precision Agriculture* **10**, 459–470 (2009).
- [95] Ares, J., Bertiller, M. & Del Valle, H. Functional and structural landscape indicators of intensification, resilience and resistance in agroecosystems in southern Argentina based on remotely sensed data. *Landscape Ecology* **16**, 221–234 (2001).
- [96] Fuller, D. O. & Prince, S. D. Rainfall and foliar dynamics in tropical Southern Africa: Potential impacts of global climatic change on savanna vegetation. *Climatic Change* **33**, 69–96 (1996).
- [97] Pinzon, J. & Tucker, C. A Non-Stationary 1981–2012 AVHRR NDVI3g Time Series. *Remote Sensing* **6**, 6929–6960 (2014).
- [98] Rouse, J. W., Haa, R. H., Schell, J. A. & Deering, D. W. Monitoring vegetation systems in the great plains with erts. In *Third Earth Resources Technology Satellite-1 Symposium- Volume I: Technical Presentations*. (1974). This is the first ndvi paper.
- [99] Matsushita, B., Yang, W., Chen, J., Onda, Y. & Qiu, G. Sensitivity of the Enhanced Vegetation Index (EVI) and Normalized Difference Vegetation Index (NDVI) to Topographic Effects: A Case Study in High-density Cypress Forest. *Sensors* **7**, 2636–2651 (2007).
- [100] JUSTICE, C. O., TOWNSHEND, J. R. G., HOLBEN, B. N. & TUCKER, C. J. Analysis of the phenology of global vegetation using meteorological satellite data. *International Journal of Remote Sensing* **6**, 1271–1318 (1985).
- [101] Hao, C., Wu, S. & Xu, C. Comparison of some vegetation indices in seasonal information. *Chinese Geographical Science* **18**, 242–248 (2008).

- [102] Higgins, S. I., Buitenwerf, R. & Moncrieff, G. R. Defining functional biomes and monitoring their change globally. *Global Change Biology* **22**, 3583–3593 (2016).
- [103] Hijmans, R. J. *raster: Geographic Data Analysis and Modeling* (2016). URL <https://CRAN.R-project.org/package=raster>. R package version 2.8-19.
- [104] Detsch, F. *gimms: Download and Process GIMMS NDVI3g Data* (2016). URL <https://CRAN.R-project.org/package=gimms>. R package version 1.1.1.
- [105] Trzaska, S. & Schnarr, E. A review of downscaling methods for climate change projections. Tech. Rep. September (2014).
- [106] ECMWF. Era5: Fifth generation of ecmwf atmospheric reanalyses of the global climate 4. (2019). URL <https://cds.climate.copernicus.eu/cdsapp>. 2018.
- [107] ECMWF. PART I: OBSERVATIONS. In *IFS Documentation CY45R1*, March, 1–76 (2018).
- [108] ECMWF. Part II : DATA ASSIMILATION. In *IFS Documentation CY45R1*, vol. March, 1–103 (2018).
- [109] ECMWF. PART III: DYNAMICS AND NUMERICAL PROCEDURES. In *IFS Documentation CY45R1*, vol. March, 1–31 (2018).
- [110] ECMWF. PART IV: PHYSICAL PROCESSES. In *IFS Documentation CY45R1*, March, 1–189 (2018).
- [111] Gazol, A., Camarero, J. J., Sangüesa-Barreda, G. & Vicente-Serrano, S. M. Post-drought Resilience After Forest Die-Off: Shifts in Regeneration, Composition, Growth and Productivity. *Frontiers in Plant Science* **9**, 1–12 (2018).
- [112] Smith, A. P. *et al.* Shifts in pore connectivity from precipitation versus groundwater rewetting increases soil carbon loss after drought. *Nature Communications* **8**, 1335 (2017).
- [113] Pagotto, M. A., Roig, F. A., de Souza Ribeiro, A. & Lisi, C. S. Influence of regional rainfall and Atlantic sea surface temperature on tree-ring growth of *Poincianella pyramidalis*, semiarid forest from Brazil. *Dendrochronologia* **35**, 14–23 (2015).
- [114] Allen, C. D. *et al.* A global overview of drought and heat-induced tree mortality reveals emerging climate change risks for forests. *Forest Ecology and Management* **259**, 660–684 (2010). arXiv:1011.1669v3.
- [115] CGIAR-CSI. Nasa shuttle radar topographic mission (srtm) (2018). URL <http://srtm.csi.cgiar.org/>. 2006.
- [116] USGS. Gtopo30 – global 30 arc second elevation data (2018). URL <https://www.usgs.gov/centers/eros/science/usgs-eros-archive-digital-elevation-global-30-arc-second-elevation-gtopo30>. 2002.
- [117] *User guide to the COMPADRE Plant Matrix Database* (2017).
- [118] Jongejans, E. & de Kroon, H. Matrix Models. In *Encyclopedia of Theoretical Ecology*, 1–14.
- [119] de Kroon, H. *et al.* Elasticities: A Review of Methods and Model Limitations. *Ecology* **81**, 607–618 (2000).
- [120] Salguero-Gómez, R. Applications of the fast-slow continuum and reproductive strategy framework of plant life histories. *New Phytologist* **213**, 1618–1624 (2017).
- [121] Capdevila, P. *et al.* Life history strategies across realms : marine animals are slower but more reproductive than terrestrial ones (2018).
- [122] Healy, K., Ezard, T. H. G., Jones, R., Salguero-Gómez, R. & Buckley, Y. M. Beyond the fast-slow continuum in animal life-histories (2018).
- [123] Paniw, M., Ozgul, A. & Salguero-Gómez, R. Interactive life-history traits predict sensitivity of plants and animals to temporal autocorrelation. *Ecology Letters* **21**, 275–286 (2018).
- [124] Stott, I., Townley, S. & Hodgson, D. J. A framework for studying transient dynamics of population projection matrix models. *Ecology Letters* **14**, 959–970 (2011).
- [125] Caswell, H. *Matrix Population Models: Construction, Analysis, and Interpretation* (Sinauer Associates, Elsevier

- Inc., 2001).
- [126] GBIF. Gbif occrences (2019). URL <https://www.gbif.org/occurrence/map>.
 - [127] Rivas-Martínez, S., Rivas-Sáenz, S. & Penas, Á. Worldwide bioclimatic classification system. *Global Geobotany* **1**, 1–638 (2011).
 - [128] Bivand, R., Keitt, T. & Rowlingson, B. *rgdal: Bindings for the Geospatial Data Abstraction Library* (2016). URL <https://CRAN.R-project.org/package=rgdal>. R package version 1.4-2.
 - [129] Rivas-Martínez, S., Rivas-Sáenz, S. & Penas, Á. Computerized bioclimatic maps of the world - ombrotypes of europe and the mideast (2019). URL http://www.globalbioclimatics.org/cbm/static/conti/Europe_Ombrotypes_gb.png.
 - [130] Rivas-Martínez, S., Rivas-Sáenz, S. & Penas, Á. Computerized bioclimatic maps of the world - ombrotypes of south america (2019). URL http://www.globalbioclimatics.org/cbm/static/conti/Europe_Ombrotypes_gb.png.
 - [131] Rivas-Martínez, S., Rivas-Sáenz, S. & Penas, Á. Computerized bioclimatic maps of the world - ombrotypes of australasia (2019). URL http://www.globalbioclimatics.org/cbm/static/conti/Europe_Ombrotypes_gb.png.
 - [132] Calaway, R., Weston, S. & Tenenbaum, D. *doParallel: Foreach Parallel Adaptor for the 'parallel' Package* (2015). URL <https://CRAN.R-project.org/package=doParallel>. R package version 1.0.10.
 - [133] Onyutha, C., Tabari, H., Rutkowska, A., Nyeko-Ogiramoi, P. & Willems, P. Comparison of different statistical downscaling methods for climate change rainfall projections over the Lake Victoria basin considering CMIP3 and CMIP5. *Journal of Hydro-environment Research* **12**, 31–45 (2016).
 - [134] Maraun, D. *et al.* Precipitation downscaling under climate change: Recent developments to bridge the gap between dynamical models and the end user. *Reviews of Geophysics* **48**, 1–38 (2010).
 - [135] Fowler, H. J., Blenkinsop, S. & Tebaldi, C. Linking climate change modelling to impacts studies: recent advances in downscaling techniques for hydrological modelling. *International Journal of Climatology* **27**, 1547–1578 (2007). joc.1556.
 - [136] Hengl, T. *A Practical Guide to Geostatistical Mapping*, vol. 44 (2011). arXiv:1011.1669v3.
 - [137] Le Riche, R., Picheny, V., Meyer, A., Kim, N.-H. & Ginsbourger, D. Gears Design with Shape Uncertainties Using Controlled Monte Carlo Simulations and Kriging (2012).
 - [138] Lichtenstern, A. Kriging methods in spatial statistics. *TUM Media Online* 97 (2013).
 - [139] Goldsmith, G. R. *et al.* Spatial variation in throughfall, soil, and plant water isotopes in a temperate forest. *Ecohydrology* e2059 (2018).
 - [140] Zuur, A. F., Ieno, E. N. & Smith, G. M. *Analysing Ecological Data*, vol. 75 of *Statistics for Biology and Health* (Springer New York, New York, NY, 2007). arXiv:1305.6995v1.
 - [141] Cailleret, M. *et al.* A synthesis of radial growth patterns preceding tree mortality. *Global Change Biology* **23**, 1675–1690 (2017).
 - [142] Legendre, P. & Legendre, L. *Numerical Ecology*, vol. 24 (Elsevier Science, 1998), 2 edn.
 - [143] Sohrabi, M. M., Ryu, J. H., Abatzoglou, J. & Tracy, J. Development of Soil Moisture Drought Index to Characterize Droughts. *Journal of Hydrologic Engineering* **20**, 04015025 (2015).
 - [144] Hiemstra, P. *automap: Automatic interpolation package*. URL <https://cran.r-project.org/web/packages/automap/index.html>. R package version 1.0.-14.
 - [145] Calaway, R., Microsoft & Weston, S. *foreach: Provides Foreach Looping Construct for R*. URL <https://cran.r-project.org/web/packages/foreach/index.html>. R package version 1.4.4.
 - [146] Wickham, H. *et al.* *ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics* (2016). URL

- <https://cran.r-project.org/web/packages/ggplot2/index.html>. R package version 3.3.0.
- [147] Pierce, D. *ncdf4: Interface to Unidata netCDF (Version 4 or Earlier) Format Data Files* (2016). URL <https://cran.r-project.org/web/packages/ncdf4/index.html>. R package version 1.16.1.
- [148] Borchers, H. W. *pracma: Practical Numerical Math Functions* (2016). URL <https://cran.r-project.org/web/packages/pracma/index.html>. R package version 2.2.2.
- [149] Chamberlain, S. *et al.* *rgbif: Interface to the Global 'Biodiversity' Information Facility API* (2016). URL <https://cran.r-project.org/web/packages/rgbif/index.html>. R package version 1.2.0.
- [150] Pebesma, E. *et al.* *sp: Classes and methods for spatial data in R* (2016). URL <https://CRAN.R-project.org/package=sp>. R package version 1.3-1.
- [151] Oksanen, J. *et al.* *vegan: Community Ecology Package* (2016). URL <https://cran.r-project.org/web/packages/vegan/index.html>. R package version 2.5-4.
- [152] Dragulescu, A. A. & Arendt, C. *xlsx: Read, Write, Format Excel 2007 and Excel 97/2000/XP/2003 Files* (2016). URL <https://cran.r-project.org/web/packages/xlsx/index.html>. R package version 0.6.1.

Appendix

A.1 Project Requirements

Following these steps ensures full reproducibility of the entire analysis. Alternatively, codes can be retrieved via <https://github.com/ErikKusch/M.Sc.Thesis>.

A.1.1 R Requirements

A.1.1.1 R Packages

Table A.1: R Packages - Packages which need to be loaded into R to fully reproduce the analyses within this study.

Package	Version	Use
automap ^[144]	1.0-14	Statistical Downscaling of ERA5 5 data
doParallel ^[132]	1.0.14	Paralell processing
foreach ^[145]	1.4.4	Paralell processing
ggplot2 ^[146]	3.1.0	Data visualisation
gimms ^[104]	1.1.1	Downloading GIMMs NDVI3g data
ncdf4 ^[147]	1.16.1	Namespace for NetCDF files
pracma ^[148]	2.2.2	Detrending time series
raster ^[103]	2.8-19	Rasterising NetCDF data
rgbif ^[149]	1.2.0	Downloading floral occurence data
rgdal ^[128]	1.4-2	Loading and using shapefiles
sp ^[150]	1.3-1	Converting point data to rasterised data
vegan ^[151]	2.5-4	PCA approach for model building
xlsx ^[152]	0.6.1	Export of numeric results

Chunk 1: Installing and loading of R packages needed to reproduce the analyses of this study.

```
install.load.package <- function(x) {
  if (!require(x, character.only = TRUE))
    install.packages(x)
  require(x, character.only = TRUE)
}

package_vec <- c("automap", "doParallel", "foreach", "ggplot2", "gimms", "gridExtra",
  "ncdf4", "pracma", "raster", "rgbif", "rgdal", "sp", "vegan", "xlsx")
sapply(package_vec, install.load.package)
```

A.1.1.2 Project Directories

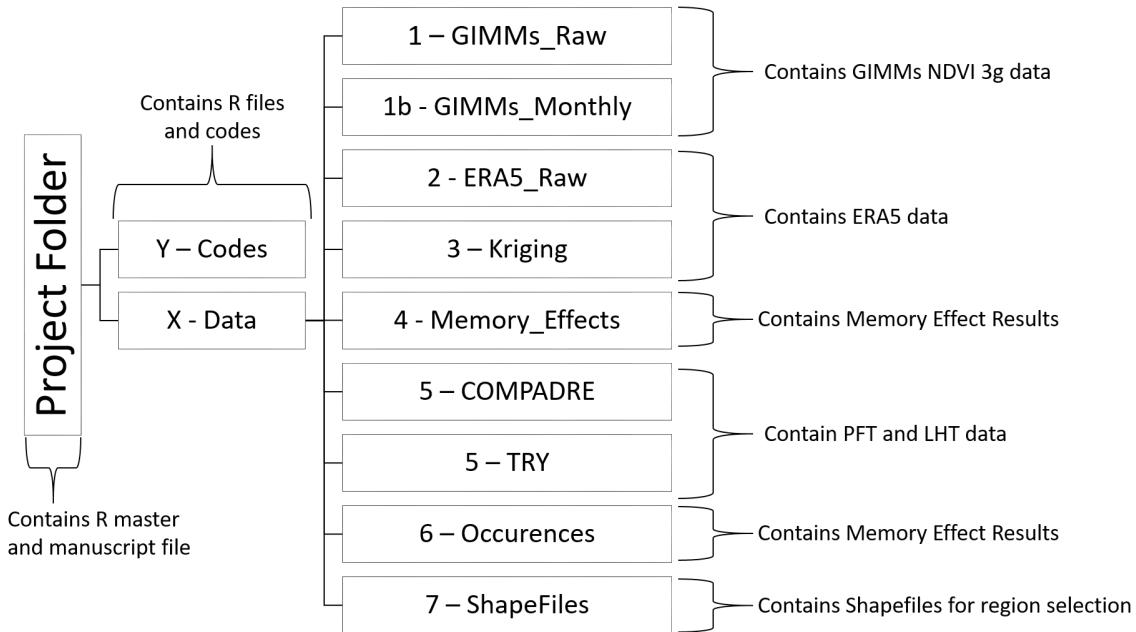


Figure A.1: Working Directories - An overview of working directories established/required by Chunk 2.

Chunk 2: Identifying and generating the folder structure used within the computational steps. A visual overview is presented in A.1.

```

mainDir <- getwd() # extract the project folder location
# WORKING DIRECTORY FOR CODES
Dir.Codes <- paste(mainDir, "/Y - Codes", sep="")
# WORKING DIRECTORY FOR DATA
Dir.Data <- paste(mainDir, "/X - Data", sep="")
# WORKING DIRECTORY FOR RAW GIMMS DATA
Dir.Gimms <- paste(Dir.Data, "/1 - GIMMs_Raw", sep="")
if(!dir.exists(Dir.Gimms)){dir.create(Dir.Gimms)}
# WORKING DIRECTORY FOR PROCESSED GIMMS DATA
Dir.Gimms.Monthly <- paste(Dir.Data, "/1b - GIMMs_Monthly", sep="")
if(!dir.exists(Dir.Gimms.Monthly)){dir.create(Dir.Gimms.Monthly)}
# WORKING DIRECTORY FOR RAW ERA5 DATA
Dir.ERA <- paste(Dir.Data, "/2 - ERA5_Raw", sep="")
# WORKING DIRECTORY FOR PROCESSED ERA5 DATA
Dir.ERA.Monthly <- paste(Dir.Data, "/2b - ERA5_Monthly", sep="")
if(!dir.exists(Dir.ERA.Monthly)){dir.create(Dir.ERA.Monthly)}
# WORKING DIRECTORY FOR KRIGING COVARIATES
Dir.KrigCov <- paste(Dir.Data, "/3 - Kriging", sep="")
# WORKING DIRECTORY FOR MEMORY EFFECT DATA
Dir.Memory <- paste(Dir.Data, "/4 - Memory_Effects", sep="")
# WORKING DIRECTORY FOR COMPADRE DATA
Dir.Compadre <- paste(Dir.Data, "/5 - COMPADRE", sep="")
# WORKING DIRECTORY FOR TRY PFT DATA
if(!dir.exists(Dir.Memory)){dir.create(Dir.Memory)}
Dir.TRY <- paste(Dir.Data, "/5 - TRY", sep="")
# WORKING DIRECTORY FOR OCCURENCE DATA
Dir.OCCs <- paste(Dir.Data, "/6 - Occurences", sep="")
if(!dir.exists(Dir.OCCs)){dir.create(Dir.OCCs)}
# WORKING DIRECTORY FOR SHAPEFILES (contains masking file for water bodies)
Dir.Mask <- paste(Dir.Data, "/7 - ShapeFiles", sep="")
  
```

A.1.1.3 R Functions for Region Selection and Raster Names

Chunk 3: User-defined functions used to optimise further functions of this analysis when (1) Limitting to study regions (RegionSelection), and (2) assigning names to model rasters (Fun_NamesRas).

```
### RegionSelection [Region, RegionFile, Extent] (selecting region and extent from
### shapefiles)
RegionSelection <- function(Region, RegionFile, Extent) {
  ## loading shapefiles
  Shapes <- readOGR(Dir.Mask, "ne_50m_admin_0_countries", verbose = FALSE)
  ## selecting region from shapefile run global analysis read user-defined extent
  ## (if applicable)
  if (Region == "Global") {
    if (is.null(Extent)) {
      area <- extent(-180, 180, -90, 90)
    } else {
      area <- Extent
    }
    location <- 1:length(Shapes) # selecting all countries contained within the shapefile
  } else {
    Where <- Region # countries to consider
    location <- NA # position vector in shapefile list
    for (i in 1:length(Where)) {
      # select region from Shapefiles
      location[i] <- which(as.vector(Shapes$NAME) == Where[i])
    }
    if (is.null(Extent)) {
      # read user-defined extent (if applicable)
      area <- extent(Shapes[location, ])
    } else {
      area <- Extent
    }
  }
  if (is.null(RegionFile)) {
    # if no file name has been specified
    RegionFile <- toString(Region) # take name of region
  }
  # returning parameters
  return(list(area, location, RegionFile))
}

### Fun_NamesRas [raster, ClimVar, ClimVar2]
# (assigning layer names to model rasters) ----
Fun_NamesRas <- function(raster, ClimVar, ClimVar2, rasiter = 1){
  names(raster) <- c(paste("Most informative", ClimVar[[rasiter]], "lag", sep=" "),
                     "Model AICs", "Model p-value", "Antecedent NDVI (c_NDVI)",
                     paste("Antecedent", ClimVar[[rasiter]], "(c_clim)", sep=" "),
                     paste("Antecedent", ClimVar2[[rasiter]], "(c_clim2)", sep=" "),
                     "Explained Variance", "Variance (NDVI)", "Variance (Shared)",
                     paste("Variance (", ClimVar[[rasiter]], ")", sep=""))
  return(raster)} # Fun_NamesRas end
```

A.1.2 ERA5 Data

A.1.2.1 Obtaining ERA5 Data

Chunk 4: Obtaining ERA5 data from the ECMW servers. Substitute the date statement to download the full data set. Python script.

```
import cdsapi

c = cdsapi.Client()
c.retrieve('reanalysis-era5-complete', {
    # do not change this!
    'class' : 'ea',
    'expver' : '1',
    'stream' : 'moda',
    'type' : 'an',
    'param' : '39.128/40.128/41.128/42.128/167.128',
    'levtype' : 'sfc',
    'date' : '19800101/19800201/19800301/19800401/19800501/19800601/19800701/19800801/19800901/19801001/19801101/
19801201/19810101/19810201/19810301/19810401/19810501/19810601/19810701/19810801/19810901/19811001/19811101/
19811201/19820101/19820201/19820301/19820401/19820501/19820601/19820701/19820801/19820901/19821001/19821101/
19821201/19830101/19830201/19830301/19830401/19830501/19830601/19830701/19830801/19830901/19831001/19831101/
19831201/19840101/19840201/19840301/19840401/19840501/19840601/19840701/19840801/19840901/19841001/19841101/
19841201/19850101/19850201/19850301/19850401/19850501/19850601/19850701/19850801/19850901/19851001/19851101/
19851201/19860101/19860201/19860301/19860401/19860501/19860601/19860701/19860801/19860901/19861001/19861101/
19861201/19870101/19870201/19870301/19870401/19870501/19870601/19870701/19870801/19870901/19871001/19871101/
19871201/19880101/19880201/19880301/19880401/19880501/19880601/19880701/19880801/19880901/19881001/19881101/
19881201/19890101/19890201/19890301/19890401/19890501/19890601/19890701/19890801/19890901/19891001/19891101/19891201',
    'decade' : '1980',
}, 'All_1980s.grib')
```

A.1.2.2 Pre-Processing ERA5 Data

Chunk 5: Processing ERA5 data from local downloads to merge by time step. Bash script.

```
#!/bin/bash

. /opt/.profiles/cdo.bash

# Convert files to netcdf format
cdo -f nc setgridtype,regular 'All_1980s.grib' 'All_1980s.nc'
cdo -f nc setgridtype,regular 'All_1990s.grib' 'All_1990s.nc'
cdo -f nc setgridtype,regular 'All_2000s.grib' 'All_2000s.nc'
cdo -f nc setgridtype,regular 'All_2010s.grib' 'All_2010s.nc'

# Change variable names to identifiable ones
cdo chname,var167,Tair,var39,Qsoil1,var40,Qsoil2,var41,Qsoil3,var42,Qsoil4 'All_1980s.nc' 'All_1980s_renamed.nc'
cdo chname,var167,Tair,var39,Qsoil1,var40,Qsoil2,var41,Qsoil3,var42,Qsoil4 'All_1990s.nc' 'All_1990s_renamed.nc'
cdo chname,var167,Tair,var39,Qsoil1,var40,Qsoil2,var41,Qsoil3,var42,Qsoil4 'All_2000s.nc' 'All_2000s_renamed.nc'
cdo chname,var167,Tair,var39,Qsoil1,var40,Qsoil2,var41,Qsoil3,var42,Qsoil4 'All_2010s.nc' 'All_2010s_renamed.nc'

# Merge the files over time
cdo -r mergetime `ls *_renamed.nc` 'All_variables_1980_2016.nc'

# Cleanup
rm *renamed.nc
```

Chunk 6: Regridding ERA5 data to fit GIMMS and HWSD extents. Matlab script.

```
clear
clc
```

```

fn          = 'All_variables_1980_2016.nc';
ncin        = netcdf.open(fn,'NC_NOWRITE');
lon_orig    = double(netcdf.getVar(ncin,0));
lat         = double(netcdf.getVar(ncin,1));
time        = double(netcdf.getVar(ncin,4));
Tair_orig(:,:,:) = double(netcdf.getVar(ncin,8));
Qsoil1_orig(:,:, :) = double(squeeze(netcdf.getVar(ncin,5)));
Qsoil1_orig(:,:, :) = double(squeeze(netcdf.getVar(ncin,5)));
Qsoil1_orig(:,:, :) = double(squeeze(netcdf.getVar(ncin,5)));
Qsoil1_orig(:,:, :) = double(squeeze(netcdf.getVar(ncin,5)));
netcdf.close(ncin)

lon_orig(lon_orig>180)=lon_orig(lon_orig>180)-360; % Convert longitude to the -180 to 180 reference

% Re-organise by longitude
[lon,I]    = sort(lon_orig);
Qsoil1 = Qsoil1_orig(I,:,:);
Qsoil2 = Qsoil2_orig(I,:,:);
Qsoil3 = Qsoil3_orig(I,:,:);
Qsoil4 = Qsoil4_orig(I,:,:);
Tair   = Tair_orig(I,:,:);

clear Qsoil1_orig Qsoil2_orig Qsoil3_orig Qsoil4_orig Tair_orig
%%
fn = ['Qsoil1_TrainingResolution.nc'];
[lon_dim,lat_dim,tim_dim] = size(Tair);
ncout = netcdf.create(fn,'CLOBBER');
lonID = netcdf.defDim(ncout,'lon',lon_dim);
latID = netcdf.defDim(ncout,'lat',lat_dim);
timID = netcdf.defDim(ncout,'time',tim_dim);
varid1 = netcdf.defVar(ncout,'lon',      'nc_float',[lonID]); %#ok<*NBRACK>
varid2 = netcdf.defVar(ncout,'lat',      'nc_float',[latID]);
varid3 = netcdf.defVar(ncout,'time',     'nc_float',[timID]);
varid5 = netcdf.defVar(ncout,'Qsoil1',   'nc_float',[lonID,latID,timID]);
% Put the attributes for the lon dimension
netcdf.putAtt(ncout,varid1,'units','degrees_east');
netcdf.putAtt(ncout,varid1,'standard_name','longitude');
% Put the attributes for the lat dimension
netcdf.putAtt(ncout,varid2,'units','degrees_north');
netcdf.putAtt(ncout,varid2,'standard_name','latitude');
% Put the attributes for the Time
netcdf.putAtt(ncout,varid3,'units','hours since 2008-01-01 06:00:00');
netcdf.putAtt(ncout,varid3,'calendar','proleptic_gregorian');
netcdf.putAtt(ncout,varid3,'standard_name','time');
netcdf.endDef(ncout)
netcdf.putVar(ncout,varid1, lon)
netcdf.putVar(ncout,varid2, lat)
netcdf.putVar(ncout,varid3, time)
netcdf.putVar(ncout,varid5, Qsoil1)
netcdf.close(ncout)

fn = ['Qsoil2_TrainingResolution.nc'];
[lon_dim,lat_dim,tim_dim] = size(Tair);
ncout = netcdf.create(fn,'CLOBBER');
lonID = netcdf.defDim(ncout,'lon',lon_dim);
latID = netcdf.defDim(ncout,'lat',lat_dim);
timID = netcdf.defDim(ncout,'time',tim_dim);

```

```

varid1 = netcdf.defVar(ncout,'lon',      'nc_float',[lonID]); %#ok<*NBRAK>
varid2 = netcdf.defVar(ncout,'lat',      'nc_float',[latID]);
varid3 = netcdf.defVar(ncout,'time',     'nc_float',[timID]);
varid5 = netcdf.defVar(ncout,'Qsoil2',   'nc_float',[lonID,latID,timID]);
% Put the attributes for the lon dimension
netcdf.putAtt(ncout,varid1,'units','degrees_east');
netcdf.putAtt(ncout,varid1,'standard_name','longitude');
% Put the attributes for the lat dimension
netcdf.putAtt(ncout,varid2,'units','degrees_north');
netcdf.putAtt(ncout,varid2,'standard_name','latitude');
% Put the attributes for the Time
netcdf.putAtt(ncout,varid3,'units','hours since 2008-01-01 06:00:00');
netcdf.putAtt(ncout,varid3,'calendar','proleptic_gregorian');
netcdf.putAtt(ncout,varid3,'standard_name','time');
netcdf.endDef(ncout)
netcdf.putVar(ncout,varid1, lon)
netcdf.putVar(ncout,varid2, lat)
netcdf.putVar(ncout,varid3, time)
netcdf.putVar(ncout,varid5, Qsoil2)
netcdf.close(ncout)

fn = ['Qsoil3_TrainingResolution.nc'];
[lon_dim,lat_dim,tim_dim] = size(Tair);
ncout = netcdf.create(fn,'CLOBBER');
lonID = netcdf.defDim(ncout,'lon',lon_dim);
latID = netcdf.defDim(ncout,'lat',lat_dim);
timID = netcdf.defDim(ncout,'time',tim_dim);
varid1 = netcdf.defVar(ncout,'lon',      'nc_float',[lonID]); %#ok<*NBRAK>
varid2 = netcdf.defVar(ncout,'lat',      'nc_float',[latID]);
varid3 = netcdf.defVar(ncout,'time',     'nc_float',[timID]);
varid5 = netcdf.defVar(ncout,'Qsoil3',   'nc_float',[lonID,latID,timID]);
% Put the attributes for the lon dimension
netcdf.putAtt(ncout,varid1,'units','degrees_east');
netcdf.putAtt(ncout,varid1,'standard_name','longitude');
% Put the attributes for the lat dimension
netcdf.putAtt(ncout,varid2,'units','degrees_north');
netcdf.putAtt(ncout,varid2,'standard_name','latitude');
% Put the attributes for the Time
netcdf.putAtt(ncout,varid3,'units','hours since 2008-01-01 06:00:00');
netcdf.putAtt(ncout,varid3,'calendar','proleptic_gregorian');
netcdf.putAtt(ncout,varid3,'standard_name','time');
netcdf.endDef(ncout)
netcdf.putVar(ncout,varid1, lon)
netcdf.putVar(ncout,varid2, lat)
netcdf.putVar(ncout,varid3, time)
netcdf.putVar(ncout,varid5, Qsoil3)
netcdf.close(ncout)

fn = ['Qsoil4_TrainingResolution.nc'];
[lon_dim,lat_dim,tim_dim] = size(Tair);
ncout = netcdf.create(fn,'CLOBBER');
lonID = netcdf.defDim(ncout,'lon',lon_dim);
latID = netcdf.defDim(ncout,'lat',lat_dim);
timID = netcdf.defDim(ncout,'time',tim_dim);
varid1 = netcdf.defVar(ncout,'lon',      'nc_float',[lonID]); %#ok<*NBRAK>
varid2 = netcdf.defVar(ncout,'lat',      'nc_float',[latID]);
varid3 = netcdf.defVar(ncout,'time',     'nc_float',[timID]);
varid5 = netcdf.defVar(ncout,'Qsoil4',   'nc_float',[lonID,latID,timID]);

```

```
% Put the attributes for the lon dimension
netcdf.putAtt(ncout,varid1,'units','degrees_east');
netcdf.putAtt(ncout,varid1,'standard_name','longitude');

% Put the attributes for the lat dimension
netcdf.putAtt(ncout,varid2,'units','degrees_north');
netcdf.putAtt(ncout,varid2,'standard_name','latitude');

% Put the attributes for the Time
netcdf.putAtt(ncout,varid3,'units','hours since 2008-01-01 06:00:00');
netcdf.putAtt(ncout,varid3,'calendar','proleptic_gregorian');
netcdf.putAtt(ncout,varid3,'standard_name','time');

netcdf.endDef(ncout)

netcdf.putVar(ncout,varid1, lon)
netcdf.putVar(ncout,varid2, lat)
netcdf.putVar(ncout,varid3, time)
netcdf.putVar(ncout,varid5, Qsoil4)
netcdf.close(ncout)

fn = ['Tair_TrainingResolution.nc'];
[lon_dim,lat_dim,tim_dim] = size(Tair);
ncout = netcdf.create(fn,'CLOBBER');

lonID = netcdf.defDim(ncout,'lon',lon_dim);
latID = netcdf.defDim(ncout,'lat',lat_dim);
timID = netcdf.defDim(ncout,'time',tim_dim);

varid1 = netcdf.defVar(ncout,'lon','nc_float',[lonID]); %#ok<*NBRAK>
varid2 = netcdf.defVar(ncout,'lat','nc_float',[latID]);
varid3 = netcdf.defVar(ncout,'time','nc_float',[timID]);
varid5 = netcdf.defVar(ncout,'Tair','nc_float',[lonID,latID,timID,ensID]);

% Put the attributes for the lon dimension
netcdf.putAtt(ncout,varid1,'units','degrees_east');
netcdf.putAtt(ncout,varid1,'standard_name','longitude');

% Put the attributes for the lat dimension
netcdf.putAtt(ncout,varid2,'units','degrees_north');
netcdf.putAtt(ncout,varid2,'standard_name','latitude');

% Put the attributes for the Time
netcdf.putAtt(ncout,varid3,'units','hours since 2008-01-01 06:00:00');
netcdf.putAtt(ncout,varid3,'calendar','proleptic_gregorian');
netcdf.putAtt(ncout,varid3,'standard_name','time');

netcdf.endDef(ncout)

netcdf.putVar(ncout,varid1, lon)
netcdf.putVar(ncout,varid2, lat)
netcdf.putVar(ncout,varid3, time)
netcdf.putVar(ncout,varid5, Tair)
netcdf.close(ncout)
```

A.2 Data

A.2.1 HWSD Data

Table A.2: HWSD Variables and Explanations - HWSD variables used within this study and their explanations. Slope aspects and inclines are recorded as number of 3 arc-second cells falling into 5 minute cells

HWSD Variable	Explanation
<i>Elevation</i>	Altitude in metres as measured from sea-level
<i>Slope_aspect_N</i>	(0°; 45°]; (315°; 360°]
<i>Slope_aspect_E</i>	(45°; 135°]
<i>Slope_aspect_S</i>	(135°; 225°]
<i>Slope_aspect_W</i>	(225°; 315°]
<i>Slope_aspect_U</i>	Slope undefined or slope incline is less than 2%
<i>Slopes1</i>	0% ≤ incline ≤ 0.5%
<i>Slopes2</i>	0.5% ≤ incline ≤ 2%
<i>Slopes3</i>	2% ≤ incline ≤ 5%
<i>Slopes4</i>	5% ≤ incline ≤ 10%
<i>Slopes5</i>	10% ≤ incline ≤ 15%
<i>Slopes6</i>	15% ≤ incline ≤ 30%
<i>Slopes7</i>	30% ≤ incline ≤ 45%
<i>Slopes8</i>	Incline > 45%

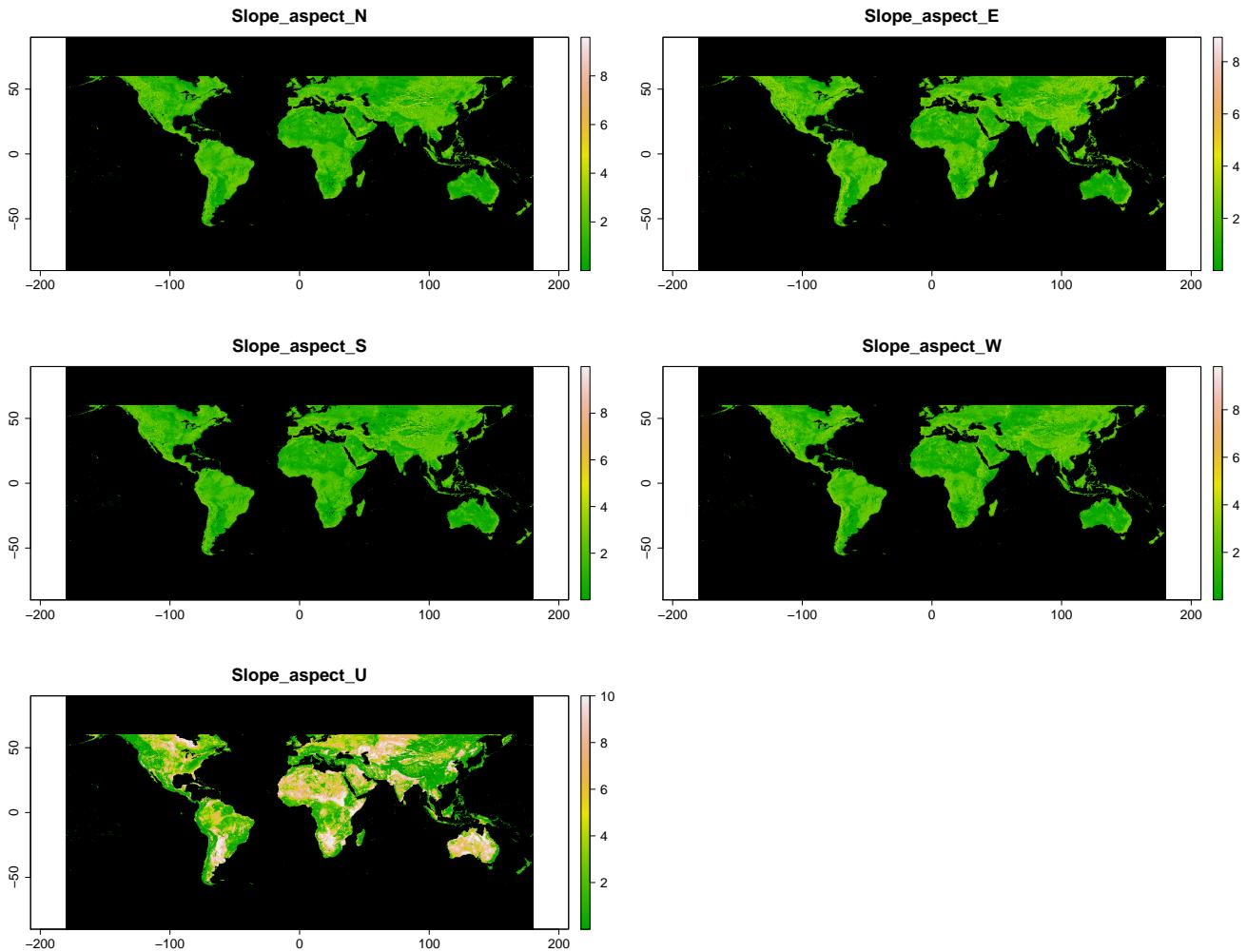


Figure A.2: HWSD slope aspect data - HWSD data at GIMMS resolution divided by 1000 for displaying. Figure established via Chunk 18.

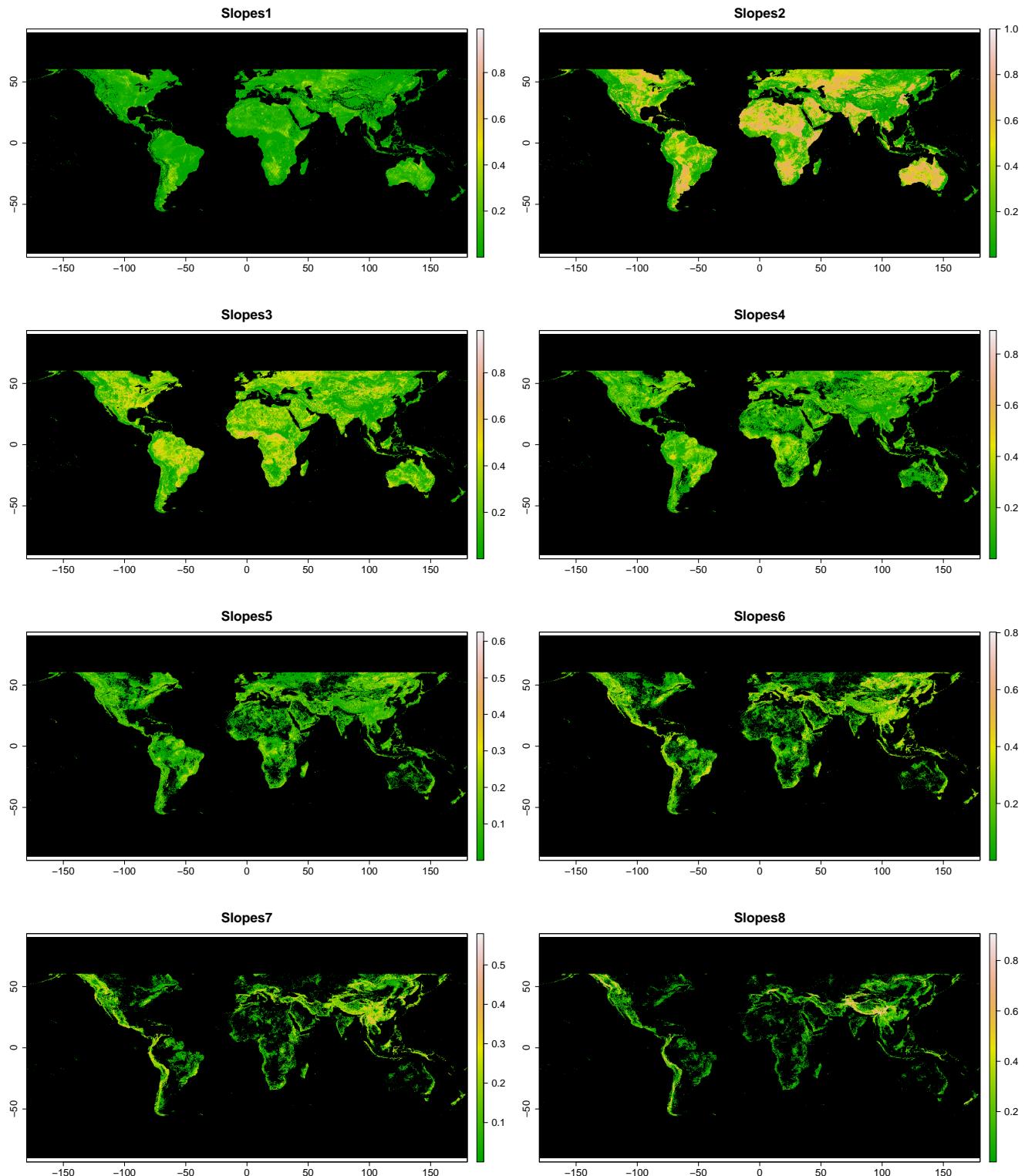


Figure A.3: HWSD slope incline data - HWSD data at GIMMS resolution. Data has been divided by 1000 for displaying. Figure established via Chunk 19.

A.2.2 Study Regions

A.2.2.1 Iberian Region

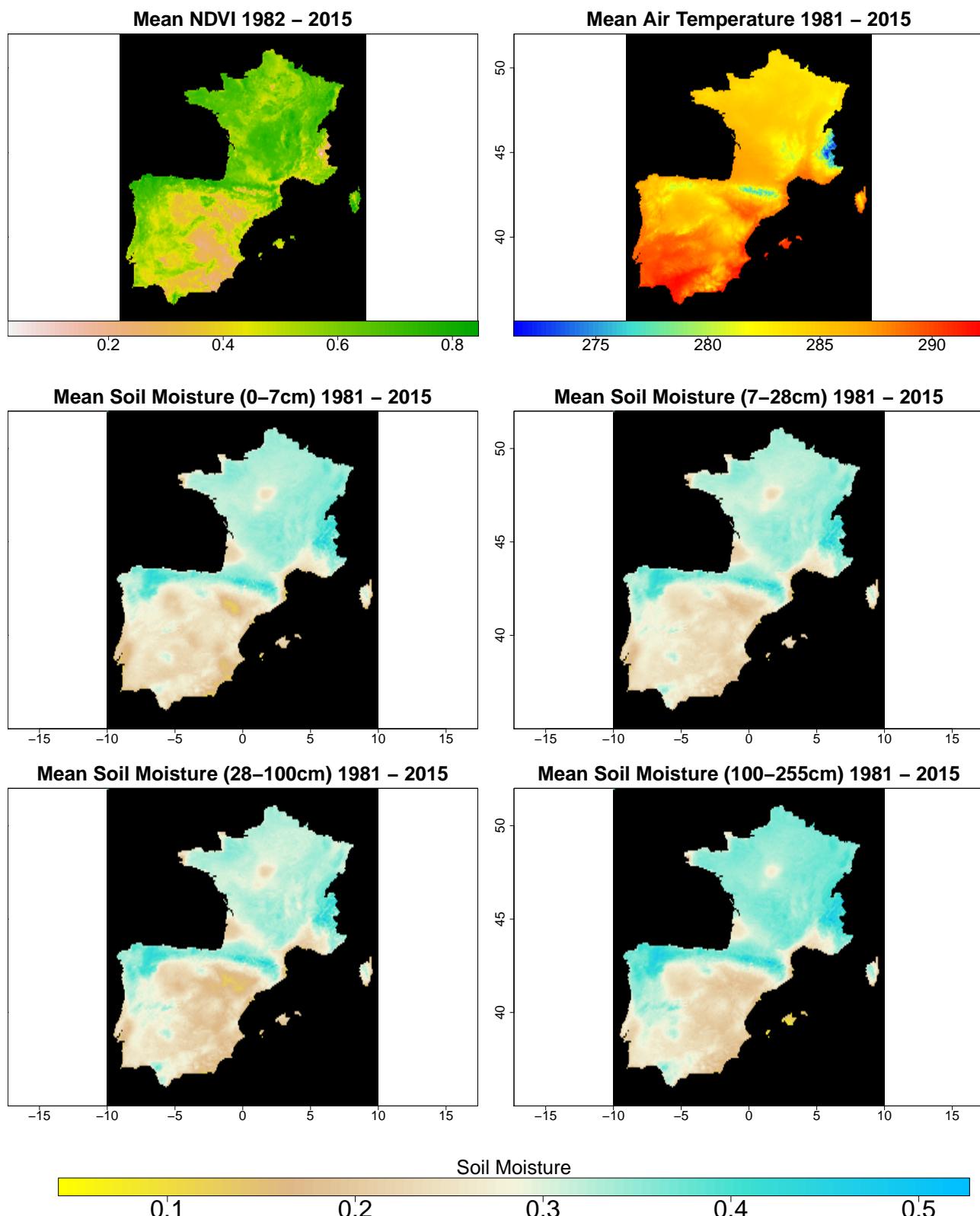


Figure A.4: Data Overview (Iberian Region) - Data required for identification of vegetation memory: (1) GIMMS NDVI 3g; and (2) ERA5: Tair, Qsoil1, Qsoil2, Qsoil3, Qsoil4. Figure established via Chunk 23.

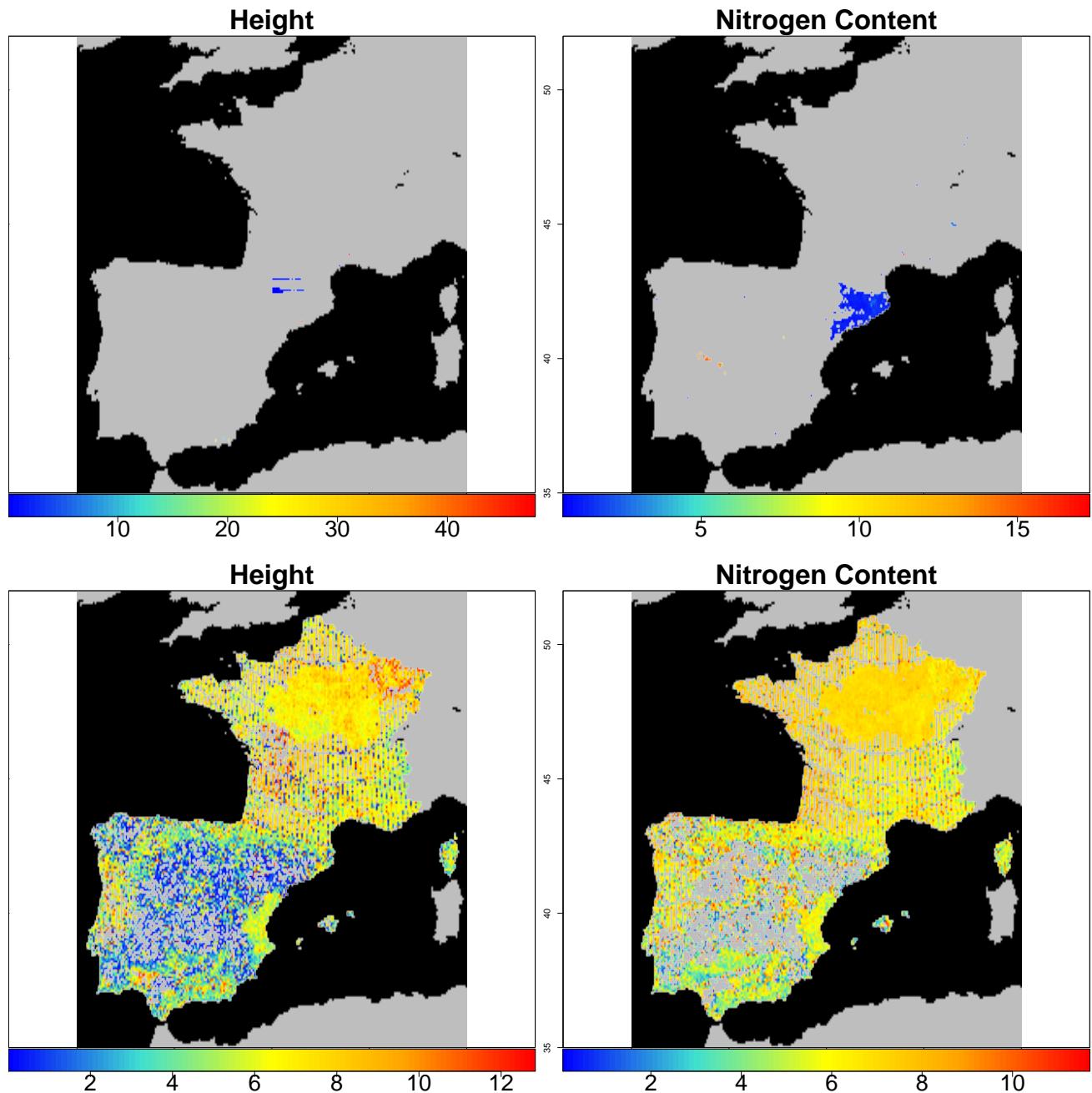


Figure A.5: TRY Data (Iberian Region) - TRY PFT data across the Iberian Region: (1) Geo-referenced records (upper row), and (2) Extrapolated species-specific mean PFT records (lower row). Figure established via Chunk 24.

A.2.2.2 Caatinga, Brazil

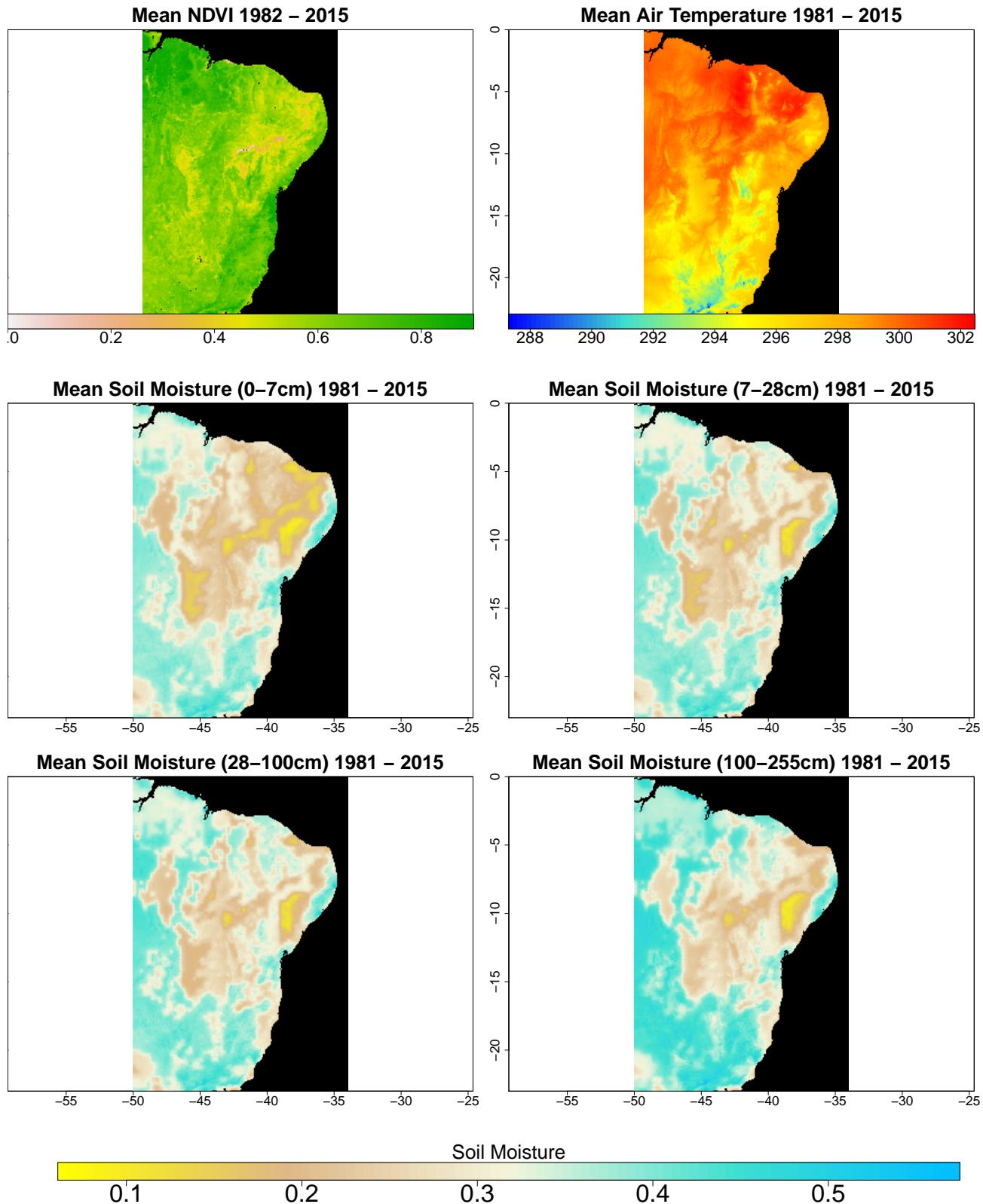


Figure A.6: Data Overview (Caatinga, Brazil) - Data required for identification of vegetation memory: (1) GIMMS NDVI 3g; and (2) ERA5: Tair, Qsoil1, Qsoil2, Qsoil3, Qsoil4. Figure established via Chunk 23.

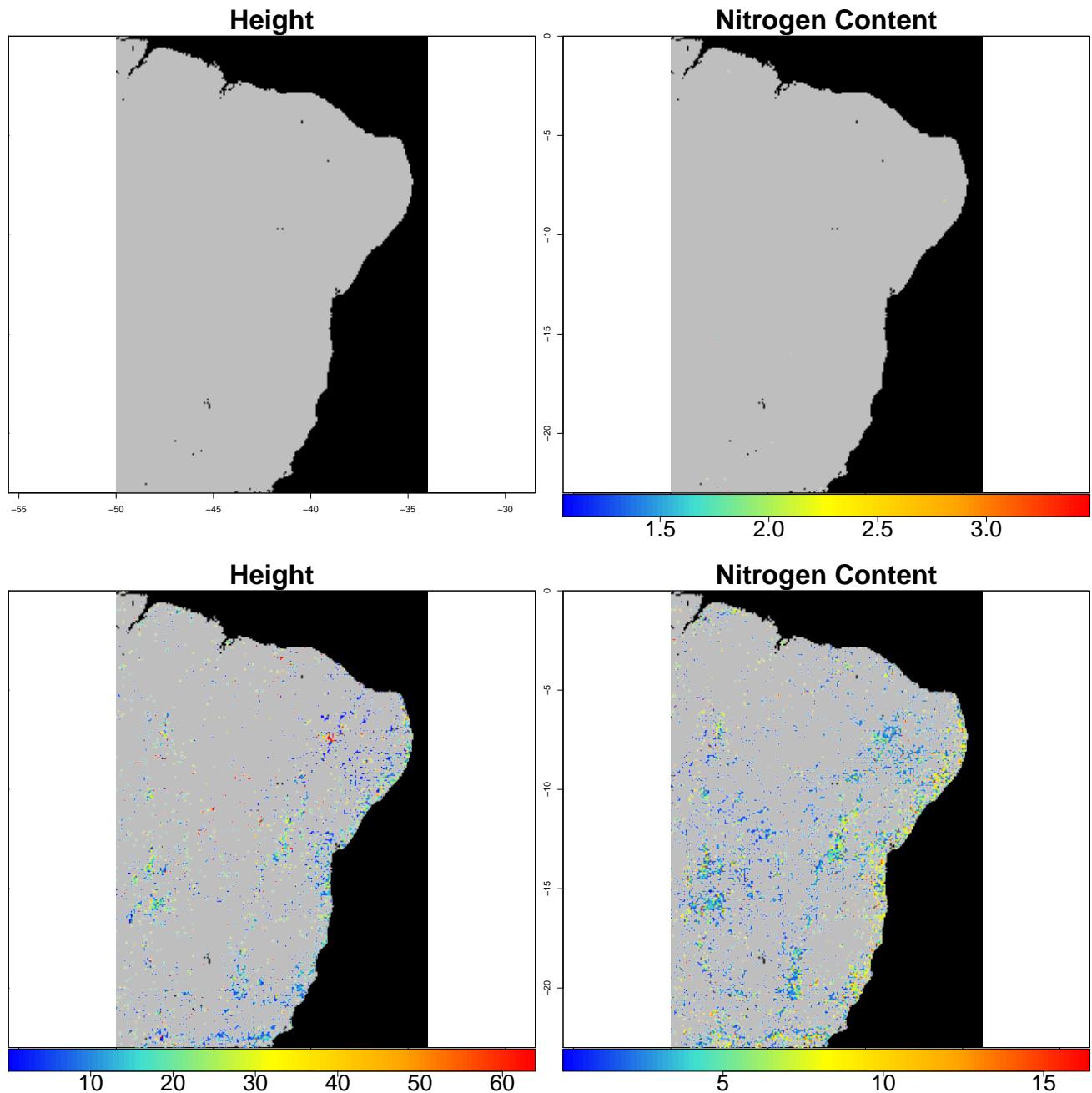


Figure A.7: TRY Data (Caatinga, Brazil) - TRY PFT data across Caatinga, Brazil: (1) Geo-referenced records (upper row), and (2) Extrapolated sepcies-specific mean PFT records (lower row). Figure established via Chunk 24.

A.2.2.3 Australia

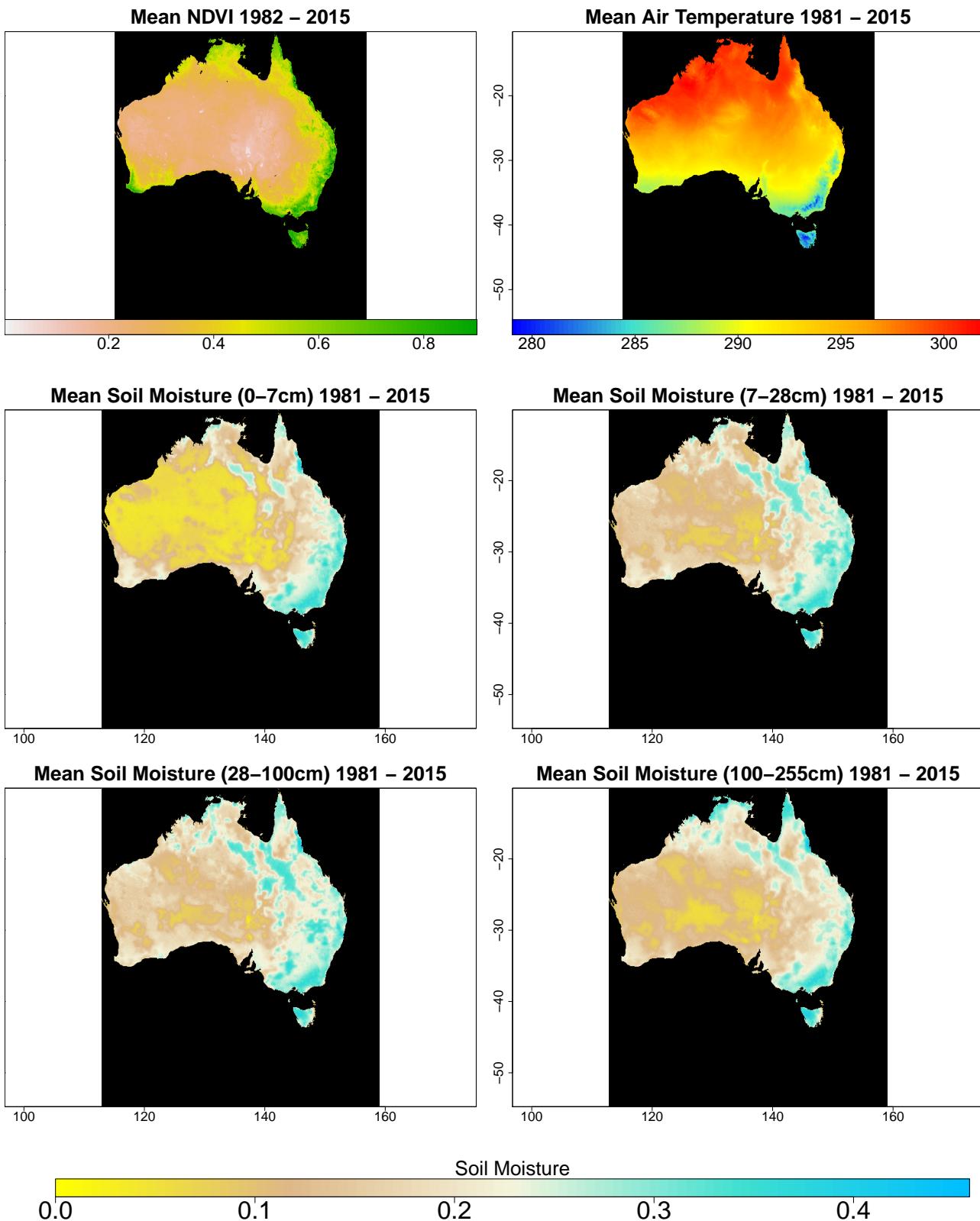


Figure A.8: Data Overview (Australia) - Data required for identification of vegetation memory: (1) GIMMS NDVI 3g; and (2) ERA5: Tair, Qsoil1, Qsoil2, Qsoil3, Qsoil4. Figure established via Chunk 23.

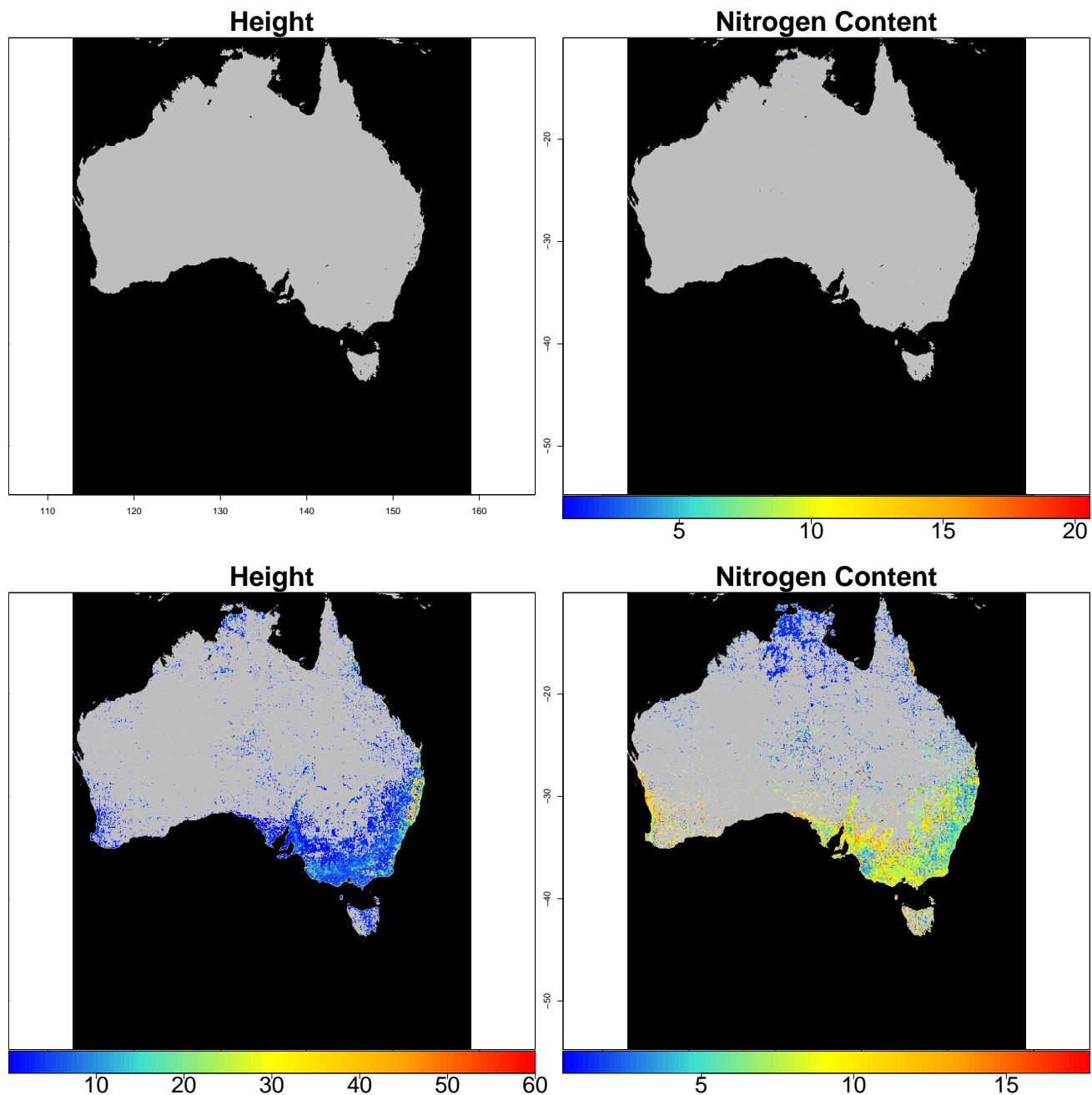


Figure A.9: TRY Data (Australia) - TRY PFT data across Australia: (1) Geo-referenced records (upper row), and (2) Extrapolated species-specific mean PFT records (lower row). Figure established via Chunk 24.

A.3 R Analyses Codes

A.3.1 Master File

Chunk 7: Running the entire analysis once data has been allocated to directories. S0a_Packages.R can be found in Chunk 1, S0b_Directories.R is located in Chunk 2 whilst S0c_Functions.R is contained in Chunk 3. S1_GIMMs.R, S2_ERA5.R, S3_VegetationMemory.R, S4_PFTs.R, and S4_COMPADRE.R are included in Chunk 8, Chunk 9, Chunk 10, Chunk 11, and Chunk 12, respectively.

```

rm(list = ls()) # clearing environment
#####----- PACKAGES -----
source("Y - Codes/S0a_Packages.R") # loading packages
#####----- DIRECTORIES -----
source("Y - Codes/S0b_Directories.R") # setting directories
#####----- FUNCTIONS -----
source("Y - Codes/S0c_Functions.R") # Loading miscellaneous functions
#####----- VARIABLE VECTORS -----
ModVars <- c("Tair", "Qsoil1", "Qsoil2", "Qsoil3", "Qsoil4")
ClimVars = list("Qsoil1_mean", "Qsoil2_mean", "Qsoil3_mean", "Qsoil4_mean")
ClimVars2 = list("Tair_mean", "Tair_mean", "Tair_mean", "Tair_mean")
#####----- FUNCTIONS -----
#####----- Fun_Vegetation [Regions, RegionFiles, Extents, From, To, Lags, Cores]
# (selecting and preparing data, and calculating vegetation memory) -----
Fun_Vegetation <- function(Regions, RegionFiles, Extents, From, To, Lags, Cores) {
  FromY <- (From - ceiling(1/12 * max(Lags))) # Figuring out real start year after factoring in lags
  ##### GIMMs NDVI -----
  print("+++++++++++++++++++++++++++++++++++++")
  print("HANDLING GIMMs NDVI DATA")
  print("+++++++++++++++++++++++++++++++++++++")
  setwd(mainDir)
  source(paste(Dir.Codes, "S1_GIMMs.R", sep="/"))
  # Download NDVI data for full years (from, to), turn it into monthly composite
  # rasters checking if all files are already there and not running this function
  # if they are
  froms <- c(1982, 1986, 1991, 1996, 2001, 2006, 2011)
  tos <- c(1985, 1990, 1995, 2000, 2005, 2010, 2015)
  for(RasGimms in 1:length(tos)){
    if(file.exists(paste(Dir.Gimms.Monthly, "/GlobalNDVI_", froms[RasGimms], tos[RasGimms], ".nc", sep=""))){
      print(paste("Global NDVI raster from", froms[RasGimms], "to", tos[RasGimms], "has already been established."))
      next()
    }else{
      RasterGIMMs(from = froms[RasGimms], to = tos[RasGimms])
    }
  }
  # Load composite NDVI data and limit to extent of a study region saving the
  # resulting data
  for (CombineRun in 1:length(Regions)) {
    # Checking if this particular data has been produced already
    if (file.exists(paste(Dir.Gimms.Monthly, "/NDVI_", RegionFiles[[CombineRun]],
                         ".nc", sep = ""))) {
      print(paste("NDVI raster stack already cropped for: ", RegionFiles[[CombineRun]],
                 sep = ""))
      (next)()
    }
    CombineCDFs(Region = Regions[[CombineRun]], RegionFile = RegionFiles[[CombineRun]],
                Extent = Extents[[CombineRun]])
  } # CombineCDFs
  ##### ERA5 -----
  print("+++++++++++++++++++++++++++++++++++++")

```

```

print("HANDLING ERA5 DATA")
print("+" + "-" * 60 + "+")
setwd(mainDir)
source(paste(Dir.Codes, "S2_ERA5.R", sep="/"))
# kriging ERA5 variable data across study regions for selected time period
# parallel
if (Cores > 1) {
  cl <- makeCluster(Cores) # Assuming X node cluster
  registerDoParallel(cl) # Register cores
  for (KrigRegion in 1:length(Regions)) {
    # looping over regions
    print("+" + "-" * 60 + "+")
    print(paste("Kriging ERA5 ", toString(ModVars), " data from across ",
               RegionFiles[[KrigRegion]], sep = ""))
    foreach(Krigrun = 1:length(ModVars)) %dopar% {
      # looping over variables
      source(paste(Dir.Codes, "S0a_Packages.R", sep="/")) # load packages to each core
      source(paste(Dir.Codes, "S0b_Directories.R", sep="/")) # set packages for each core
      source(paste(Dir.Codes, "S0c_Functions.R", sep="/")) # Loading miscellaneous functions
      source(paste(Dir.Codes, "S2_ERA5.R", sep="/")) # source function for each core
      ModVars <- c("Tair", "Qsoil1", "Qsoil2", "Qsoil3", "Qsoil4")
      # Checking if this particular data has been kriged already
      if (!file.exists(paste(Dir.ERA.Monthly, "/", ModVars[[Krigrun]],
                            "_mean_", RegionFiles[[KrigRegion]], "_", 1, FromY, "_", 12, To,
                            ".nc", sep = "")))
      {
        sapply(package_vec, install.load.package)
        RasterEra5(Variable = ModVars[[Krigrun]], Region = Regions[[KrigRegion]],
                   RegionFile = RegionFiles[[KrigRegion]], Extent = Extents[[KrigRegion]],
                   FromY = FromY, FromM = 1, ToY = To, ToM = 12, Temporary = "Keep")
      } # check if already kriged
    } # parallel run
  } # Region-loop
  stopCluster(cl) # stop cluster
} else {
  # non-parallel looping over regions looping over variables Checking if this
  # particular data has been kriged already
  for (KrigRegion in 1:length(Regions)) {
    for (Krigrun in 1:length(ModVars)) {
      if (file.exists(paste(Dir.ERA.Monthly, "/", ModVars[[Krigrun]], "_mean_",
                            RegionFiles[[KrigRegion]], "_", 1, FromY, "_", 12, To, ".nc", sep = ""))) {
        print(paste(ModVars[[Krigrun]], " data already kriged for: ", RegionFiles[[KrigRegion]],
                   sep = ""))
        (next)()
      }
      RasterEra5(Variable = ModVars[[Krigrun]], Region = Regions[[KrigRegion]],
                 RegionFile = RegionFiles[[KrigRegion]], Extent = Extents[[KrigRegion]],
                 FromY = FromY, FromM = 1, ToY = To, ToM = 12, Temporary = "Keep")
    } # Variable-loop
  } # region-loop
} # RasterEra5 function
##### VEGETATION MEMORY -----
print("+" + "-" * 60 + "+")
print("IDENTIFYING VEGETATION MEMORY")
print("+" + "-" * 60 + "+")
setwd(mainDir)
source(paste(Dir.Codes, "S3_VegetationMemory.R", sep="/"))
# Calculating vegetation memory

```

```

`%nin%` = Negate(`%in%`) # create a 'not in' statement
if (Cores > 1) {
  # parallel
  cl <- makeCluster(Cores) # Assuming X node cluster
  registerDoParallel(cl) # registering cores
  for (MemReg in 1:length(Regions)) {
    # looping over regions
    print("#####")
    print(paste("Calculating vegetation memory according to ", toString(ModVars),
                " across ", RegionFiles[[MemReg]], sep = ""))
    foreach(Memrun = 2:length(ModVars)) %dopar% {
      # looping over variables
      source(paste(Dir.Codes, "S0a_Packages.R", sep="/")) # load packages to each core
      source(paste(Dir.Codes, "S0b_Directories.R", sep="/")) # register directories with each core
      source(paste(Dir.Codes, "S3_VegetationMemory.R", sep="/")) # source functions for each core
      ModVars <- c("Tair", "Qsoil1", "Qsoil2", "Qsoil3", "Qsoil4")
      # check if already computed
      if (paste(RegionFiles[[MemReg]], "_Tair_mean-", ModVars[Memrun],
                "_mean", paste(Lags, collapse = "_"), "_", FromY, "-",
                To, ".nc",
                sep = "") %nin% list.files(Dir.Memory)) {
        VegMem(ClimVar = paste(ModVars[Memrun], "_mean", sep = ""),
               ClimVar2 = "Tair_mean",
               Region = RegionFiles[[MemReg]], Cumlags = Lags, FromY = FromY,
               ToY = To)
      }
    } # parallel loop
  } # region-loop
  stopCluster(cl) # stop cluster
} else {
  # non-parallel looping over regions looping over coefficients of soil layers
  # check if already computed
  for (MemReg in 1:length(Regions)) {
    for (Memrun in 2:length(ModVars)) {
      if (paste(RegionFiles[[MemReg]], "_Tair_mean-", ModVars[Memrun],
                "_mean", paste(Lags, collapse = "_"), "_", FromY, "-",
                To, ".nc",
                sep = "") %nin% list.files(Dir.Memory)) {
        VegMem(ClimVar = paste(ModVars[Memrun], "_mean", sep = ""),
               ClimVar2 = "Tair_mean",
               Region = RegionFiles[[MemReg]], Cumlags = Lags, FromY = FromY,
               ToY = To)
      } else {
        print(paste("Vegetation memory already computed for", ModVars[Memrun],
                    " across:", RegionFiles[[MemReg]], sep = " "))
      }
    } # memory-loop
  } # region-loop
} # VegMem function
# scaling coefficients per region to be represented on fixed scales looping over
# regions
for (MemReg in 1:length(Regions)) {
  CoeffScaling(ClimVar = ClimVars, ClimVar2 = ClimVars2, Region = list(RegionFiles[[MemReg]],
                        RegionFiles[[MemReg]],
                        RegionFiles[[MemReg]],
                        RegionFiles[[MemReg]]),
               Cumlags = list(Lags, Lags, Lags, Lags), FromY = FromY, ToY = To, UAbs = TRUE)
} # CoeffScaling function
} # Fun_Vegetation
##### Fun_PFTs [Regions, RegionFiles, Extents, From, To, Occ]
# (aggregating PFT data, downloading species occurrences, building PFT rasters)
# ----

```

```

Fun_PFTs <- function(Regions, RegionFiles, Extents, From, To, Occ) {
  source(paste(Dir.Codes, "S4_PFTs.R", sep="/"))
  print("+++++++++++++++++++++++++++++++++++++")
  print("CALCULATING SPECIES SPECIFIC-TRAIT MEANS")
  print("+++++++++++++++++++++++++++++++++++++")
  # calculating species-specific trait means
  if (!file.exists(paste(Dir.TRY, "/SpeciesTraits.RData", sep = ""))) {
    PFTs() # species-specific trait means
  } else {
    print("Species-specific trait means already calculated")
  }
  print("+++++++++++++++++++++++++++++++++++++")
  print("OBTAINING SPECIES OCCURENCE RECORDS FROM GBIF")
  print("+++++++++++++++++++++++++++++++++++++")
  # figuring out ISO3166 codes for selected regions
  ISO3166_df <- read.csv(paste(Dir.Mask, "/ISO3166.csv", sep = "")) # read ISO3166 country code list
  CountCodes <- ISO3166_df$ISO_3166_ALPHABET[which(ISO3166_df$Country %in% unlist(Regions))]
  CountCodes <- toString(CountCodes)
  CountCodes <- gsub(pattern = " ", " ", replacement = ";", x = CountCodes)
  # Download occurrence files from GBIF
  if (Occ == "Download") {
    GbifStat <- NULL
    # species occurrences
    Attempt <- 0
    while (is.null(GbifStat) || GbifStat != "Done") {
      if (Attempt > 0) {
        print("Encountered an error and starting the downloading of GBIF occurrence data again.
              This is usually due to issues with the GBIF server connection and you don't have
              to worry as long as your internet connection is stable.")
      }
      Attempt <- Attempt + 1
      try(GbifStat <- DistMaps(Species = "All", Years = From:To, CountCodes = CountCodes))
    }
  } else {
    print("Occurrence data will not be downloaded according to function call.")
  }
  # generate mean rasters of PFTs across specific regions looping over regions
  for (CompReg in 1:length(Regions)) {
    if (!file.exists(paste(Dir.TRY, "/TRY-", RegionFiles[[CompReg]], ".nc", sep = ""))) {
      PFTRasters(Region = Regions[[CompReg]], RegionFile = RegionFiles[[CompReg]],
                 Extent = Extents[[CompReg]], CountCodes = CountCodes)
    } else {
      print(paste("TRY data already aggregated to mean raster for region ",
                 RegionFiles[[CompReg]], sep = ""))
    }
  } # region-loop
} # Fun_PFTs
#####----- Fun_Compadre [Variables, Regions, RegionFiles, Extents]
# (selecting and preparing data, and making COMPADRE data into rasters) ----
Fun_COMPADRE <- function(Variables, Regions, RegionFiles, Extents) {
  print("+++++++++++++++++++++++++++++++++++++")
  print("COMPADRE ANALYSES")
  print("+++++++++++++++++++++++++++++++++++++")
  source(paste(Dir.Codes, "S4_COMPADRE.R", sep="/"))
  # Build rasters of compadre variables across study regions looping over Compadre
  # variables looping over regions
  for (CompVar in 1:length(Variables)) {
    for (CompReg in 1:length(Regions)) {

```

```

if (!file.exists(paste(Dir.Compadre, "/", Variables[[CompVar]], "/",
                      Variables[[CompVar]], "_", RegionFile = RegionFiles[[CompReg]], ".nc",
                      sep = ""))) {
  RasterCOMPADRE(Variable = Variables[[CompVar]], Region = Regions[[CompReg]],
                  RegionFile = RegionFiles[[CompReg]], Extent = Extents[[CompReg]])
} else {
  print(paste(Variables[[CompVar]], " already rasterised across ",
              RegionFiles[[CompReg]], sep = ""))
}
} # region-loop
} # CompVar-loop
} # Fun_Compadre
#####----- FUNCTION CALLS -----
Fun_Vegetation(Regions = list(c("Portugal", "Spain", "France", "Andorra"), "Brazil", "Australia"),
               RegionFiles = list("Iberian Region", "Caatinga", "Australia"),
               Extents = list(extent(-10,10,35,52), extent(-50,-34,-23,0), NULL),
               From = 1982, To = 2015, Lags = 0:12, Cores = 5)

Fun_PFTs(Regions = list(c("Portugal", "Spain", "France", "Andorra"), "Brazil", "Australia"),
          RegionFiles = list("Iberian Region", "Caatinga", "Australia"),
          Extents = list(extent(-10,10,35,52), extent(-50,-34,-23,0), NULL),
          From = 1982, To = 2015, Occ = "Download")

Fun_COMPADRE(Variables = list("Reactivity", "Rho", "Pi", "FastSlow"),
             Regions = list(c("Portugal", "Spain", "France", "Andorra"), "Brazil", "Australia"),
             RegionFiles = list("Iberian Region", "Caatinga", "Australia"),
             Extents = list(extent(-10,10,35,52), extent(-50,-34,-23,0), NULL))

```

A.3.2 Vegetation Memory

A.3.2.1 GIMMs Data

Chunk 8: Downloading GIMMs NDVI_{3g} data, establishing rasters of monthly composites (RasterGIMMs), and creating NDVI raster stacks for each study region (CombineCDFs).

```

setwd(Dir.Gimms)
gimms_files <- updateInventory
#####----- RasterGIMMs [from, to] -----
RasterGIMMs <- function(from, to){
  print("#####")
  print(paste("Rasterising GIMMs NDVI data from ", from, " to ", to, sep=""))
  setwd(Dir.Gimms) # set working directory to base GIMMs folder
  # PREPARING DATA---
  invisible(capture.output(gimms_files <- downloadGimms(x = as.Date(paste(from,"-01-01",sep="")), # start date
                                                               y = as.Date(paste(to,"-12-31",sep="")), # end date
                                                               dsn = Dir.Gimms))) # where to store the files

  gimms_raster <- rasterizeGimms(x = gimms_files, remove_header = TRUE) # rasterising
  indices <- monthlyIndices(gimms_files) # extract month indices from file list (should be two of each)
  gimms_raster_mvc <- monthlyComposite(gimms_raster, indices = indices) # create composites according to indices
  # Fix NDVI misbehaviours
  gimms_raster_mvc[gimms_raster_mvc<0] <- 0 # set threshold for barren land (NDVI<0)
  gimms_raster_mvc[gimms_raster_mvc>1] <- 1 # set threshold for saturated NDVI (NDVI > 1)
  # Indices
  Years <- rep(seq(from, to, 1), each = 12) # The year corresponding to each month in the stack
  names(gimms_raster_mvc) <- paste(month.abb, Years, sep = "") # create names for rasters
  # SAVING DATA---
  writeRaster(gimms_raster_mvc, paste(Dir.Gimms.Monthly, "/GlobalNDVI_", from, to, sep=""),
             overwrite=TRUE, format="CDF", varname="GIMMsNDVI")
  setwd(mainDir)})# end of RasterGIMMs-function
#####----- CombineCDFs [Region, RegionFile, Extent] -----
CombineCDFs <- function(Region, RegionFile, Extent){
  print("#####")
  print(paste("Producing cropped GIMMs NDVI raster stacks for ", RegionFile, sep=""))
  # SELECTING FILES---
  files <- list.files(Dir.Gimms.Monthly)
  files.pos <- grep("GlobalNDVI", files)
  # REGION SELECTION---
  Shapes <- readOGR(Dir.Mask,'ne_50m_admin_0_countries', verbose = FALSE)
  RegObj <- RegionSelection(Region = Region, RegionFile = RegionFile, Extent = Extent)
  area <- RegObj[[1]]
  location <- RegObj[[2]]
  RegionFile <- RegObj[[3]]
  # LOADING, CROPPING AND MASKING---
  setwd(Dir.Gimms.Monthly)
  ras <- list()
  for(i in 1:length(files.pos)){
    rasinter <- brick(files[pos[i]]) # load i-th ndvi file
    rasinter <- crop(rasinter, area) # cropping to extent
    rasinter <- mask(rasinter, Shapes[location,]) # masking via Shapefile
    ras[[i]] <- rasinter # save masked ndvi to list
  }
  ras <- brick(ras) # create one big regional ndvi raster
  # SAVING DATA---
  writeRaster(ras, paste(Dir.Gimms.Monthly, "/NDVI_", RegionFile, sep=""),
             overwrite=TRUE, format="CDF", varname="NDVI",
             longname= paste("Monthly NDVI means across ", Region, sep=""))
  setwd(mainDir)}) # CombineCDFs

```

A.3.2.2 ERA5

Chunk 9: Kriging ERA5 data from ERA5 to GIMMS resolution using HWSD covariates (table A.2) (RasterEra5).

```
#####----- PREAMBLE -----#
Variables_vec <- c("Tair_mean", "Qsoil1_mean", "Qsoil2_mean", "Qsoil3_mean", "Qsoil4_mean")
VariablesNames_vec <- c("Air Temperature", "Soil Moisture (0-7cm)", "Soil Moisture (7-28cm)",
                      "Soil Moisture (28-100cm)", "Soil Moisture (100-255cm)")
Covariates_vec <- c("Slopes1", "Slopes2", "Slopes3", "Slopes4", "Slopes5", "Slopes6", "Slopes7", "Slopes8",
                     "Slope_aspect_N", "Slope_aspect_E", "Slope_aspect_S", "Slope_aspect_W", "Slope_aspect_U",
                     "Elevation")
Variables_vec <- c("Tair_mean", "Qsoil1_mean", "Qsoil2_mean", "Qsoil3_mean", "Qsoil4_mean")
VariablesNames_vec <- c("Air Temperature", "Soil Moisture (0-7cm)", "Soil Moisture (7-28cm)",
                       "Soil Moisture (28-100cm)", "Soil Moisture (100-200cm)")
Covariates_vec <- c("Slopes1", "Slopes2", "Slopes3", "Slopes4", "Slopes5", "Slopes6", "Slopes7", "Slopes8",
                     "Slope_aspect_N", "Slope_aspect_E", "Slope_aspect_S", "Slope_aspect_W", "Slope_aspect_U",
                     "Elevation")
#####----- RasterEra5 [Variable, Region, FromY, FromM, ToY, ToM, Temporary] -----
# (selecting data, downscaling, exporting rasters) ----#
RasterEra5 <- function(Variable, Region, RegionFile, Extent, FromY, FromM, ToY, ToM, Temporary){
  VarPos <- which(Variables_vec == paste(Variable, "mean", sep="_")) # position for indexing of variable
  YearVec <- rep(1980:2015, each = 12) # Year vector to indicate months for time frame selection
  # CONSOLE MESSAGE---
  print("#####-----")
  print(paste("Kriging ERA5 ", VariablesNames_vec[VarPos], " data from ", FromM, "/", FromY, " to ", ToM, "/", ToY,
             " across ", RegionFile, sep=""))
  # FORMULAE VECTORS---
  Krig_formula <- "ERA5 ~ Slopes1+Slopes2+Slopes3+Slopes4+Slopes5+Slopes6+Slopes7+Slopes8+Slope_aspect_N+
  Slope_aspect_E+Slope_aspect_S+Slope_aspect_W+Slope_aspect_U+Elevation+
  Slopes1:Slope_aspect_N+Slopes2:Slope_aspect_N+Slopes3:Slope_aspect_N+
  Slopes4:Slope_aspect_N+Slopes5:Slope_aspect_N+Slopes6:Slope_aspect_N+
  Slopes7:Slope_aspect_N+Slopes8:Slope_aspect_N+Slopes1:Slope_aspect_S+
  Slopes2:Slope_aspect_S+Slopes3:Slope_aspect_S+Slopes4:Slope_aspect_S+
  Slopes5:Slope_aspect_S+Slopes6:Slope_aspect_S+Slopes7:Slope_aspect_S+
  Slopes8:Slope_aspect_S"
  # LOAD DATA---
  ## Era5 data
  FirstMonth <- which(YearVec == FromY)[FromM] # first month to consider
  LastMonth <- which(YearVec == ToY)[ToM] # last month to consider
  ras <- list() # create empty list for era5 raster data
  Montquence <- FirstMonth:LastMonth
  ras <- brick(paste(Dir.ERA, "/", Variable, "_TrainingResolution.nc", sep="")) # loading data
  ras <- ras[[Montquence]] # limitting to sought-after months
  extent(ras) <- c(-180,180,-90,90) # fix extent
  ## Covariates for Kriging
  Cov_coarse <- list() # create empty list
  for(c in 1:length(Covariates_vec)){ # cycle through all covariates and load the data
    Cov_coarse[[c]] <- raster(paste(Dir.KrigCov, "/Co-variates_TrainingResolution.nc", sep=""),
                               varname = Covariates_vec[c])}
  Cov_coarse <- brick(Cov_coarse) # make coarse covariate data into one big brick
  extent(Cov_coarse) <- c(-180,180,-90,90) # fix extent
  Cov_fine <- list() # create empty list
  for(c in 1:length(Covariates_vec)){ # cycle through all covariates and load the data
    Cov_fine[[c]] <- raster(paste(Dir.KrigCov, "/Co-variates_NativeResolution.nc", sep=""),
                            varname = Covariates_vec[c])}
  Cov_fine <- brick(Cov_fine) # make fine covariate data into one big brick
  # REGION SELECTION---
  Shapes <- readOGR(Dir.Mask,'ne_50m_admin_0_countries', verbose = FALSE)
  RegObj <- RegionSelection(Region = Region, RegionFile = RegionFile, Extent = Extent)
```

```

area <- RegObj[[1]]
location <- RegObj[[2]]
RegionFile <- RegObj[[3]]
# CROPPING AND MASKING---
## Era5 cropping and masking
ras <- crop(ras, area) # cropping to extent
ras <- mask(ras, Shapes[location,]) # masking via Shapefile
## Coarse covariate cropping and masking
Cov_coarse <- crop(Cov_coarse, area) # cropping to extent
Cov_coarse <- mask(Cov_coarse, Shapes[location,]) # masking via Shapefile
## Fine covariate cropping and masking
Cov_fine <- crop(Cov_fine, area) # cropping to extent
Cov_fine <- mask(Cov_fine, Shapes[location,]) # masking via Shapefile
# KRIING---
Months1 <- (ToY-FromY-1)*12 # how many months to cover just by years
Months2 <- abs(ToM-FromM+1) # how many months to cover only taking months of time frame into account
FullMonths <- Months1+Months2 # total count of months that are covered
if(FromY == ToY){# if range doesn't exceed a calendar year
  Years <- rep(YearVec[FirstMonth], Months2) }else{
  FromLeft <- 12-FromM+1 # months left in starting year
  ToCovered <- ToM # months to be covered in final year
  Years1 <- rep(FromY, FromLeft)
  Years2 <- rep(ToY, ToCovered)
  if(ToY-FromY > 1){
    Years3 <- rep((FromY+1):(ToY-1), each = 12) # months in full years
    Years <- c(Years1, Years3, Years2) }else{
    Years <- c(Years1, Years2)}}
  Months <- rep(c(1:12), length = length(Years))
  Names <- paste(month.abb, Years, sep="") # combination of month names and years
## Preparing Kriging
Dir.Temp <- paste(Dir.ERA.Monthly,"/Temp_",Variable,"_",RegionFile, sep="")
dir.create(Dir.Temp)
TempNames <- paste(rep(YearVec),
                     rep(c("01","02","03","04","05","06","07","08","09","10","11","12")),sep="_")
# figuring out where to begin with the names
if(FromM < 10){
  TempStart <- which(TempNames == paste(FromY,"_0",FromM, sep="")) }else{
  TempStart <- which(TempNames == paste(FromY,"_",FromM, sep=""))}
# figuring out where to stop with the names
if(ToM < 10){
  TempStop <- which(TempNames == paste(ToY,"_0",ToM, sep="")) }else{
  TempStop <- which(TempNames == paste(ToY,"_",ToM, sep=""))}
TempNames <- TempNames[TempStart:TempStop]
Ras_Krig <- list()
### Actual Kriging
counter <- 0
for(i in 1:length(names(ras))){
  if(paste(TempNames[i], ".nc", sep="") %in% list.files(Dir.Temp)){ # check if this file has already been produced
    print(paste(TempNames[i], "already kriged", sep=" "))
    Ras_Krig[[i]] <- raster(paste(Dir.Temp, "/", TempNames[i], ".nc", sep=""))
    next()}
  counter <- counter + 1
  T_Begin <- Sys.time()
  RasterX <- ras[[i]]# extracting raster from Era5 stack
  # Base and Covariate Coarse Data
  Origin <- as.data.frame(RasterX, xy = TRUE)
  Origin <- na.omit(Origin)
  for(c in 1:length(Covariates_vec)){
    
```

```

Cov_coarse[[c]][!is.na(RasterX) & is.na(Cov_coarse[[c]])] <- 0 # 0 cells where no info
Cov_coarse[[c]][is.na(RasterX)] <- NA # ensure same NAs
Covariate <- as.data.frame(Cov_coarse[[c]], xy = TRUE)
Covariate <- na.omit(Covariate)
Origin <- cbind(Origin, Covariate[,3])
colnames(Origin) <- c("x", "y", "ERA5", Covariates_vec)
# checking data availability
for(it_check in 1:length(colnames(Origin))){
  if(length(which(Origin[,it_check] != 0)) < 2){
    stop(paste("The native resolution data does not support kriging using the formula you have specified
      because ", colnames(Origin)[it_check], " does not contain enough data records for kriging
      to be performed across the region you have specified (", Region, ".)", " You can resolve
      this issue by either removing the interaction effects containing this variable from the
      formula or choosing a bigger study region.", sep=""))}
  }
OriginK <- Origin
gridded(OriginK) <- ~x+y
# CROPPING TARGET
Cov_fine[[1]][which(is.na(as.vector(Cov_fine[[1]])))] <- 0
Cov_fine[[1]] <- mask(Cov_fine[[1]], Shapes[location,])
Target <- as.data.frame(Cov_fine[[1]], xy = TRUE)
for(c in 2:length(Covariates_vec)){
  Cov_fine[[c]][which(is.na(as.vector(Cov_fine[[c]])))] <- 0
  Cov_fine[[c]] <- mask(Cov_fine[[c]], Shapes[location,])
  Covariate <- as.data.frame(Cov_fine[[c]], xy = TRUE)
  Target <- cbind(Target, Covariate[,3])
  colnames(Target) <- c("x", "y", Covariates_vec)
  TargetK <- Target
  gridded(TargetK) <- ~x+y
# KRIGING
invisible(capture.output(
  kriging_result <- autoKrigie(
    as.formula(Krig_formula), OriginK, TargetK, verbose = FALSE))
Krig_ras <- raster(kriging_result$krige_output)
Ras_Krig[[i]] <- Krig_ras
# writing the raster
writeRaster(Krig_ras, filename = paste(Dir.Temp, "/", TempNames[i], sep=""), overwrite=TRUE, format="CDF")
if(counter == 1){
  T_End <- Sys.time()
  Duration <- as.numeric(T_End)-as.numeric(T_Begin)
  print(paste("Calculating monthly ERA5 ", VariablesNames_vec[VarPos], " rasters from ", FromM, "/",
    FromY,
    " to ", ToM, "/", ToY, " across ", RegionFile, " should finish around: ",
    as.POSIXLT(T_Begin + Duration*(length(names(ras))-i), tz = Sys.timezone(location=TRUE)), sep=""))
  pb <- txtProgressBar(min = 0, max = length(names(ras)), style = 3)
  setTxtProgressBar(pb, i)} # kriging loop
# COMBINING KRIGED ENSEMBLES FROM MEMORY----
Ras_Krig <- brick(Ras_Krig)
ras <- Ras_Krig
# ELIMINATE KRIGING ARTIFACTS OF SOIL MOISTURE BY BOUNDING
if(Variable == "Qsoil1" | Variable == "Qsoil2" | Variable == "Qsoil3" | Variable == "Qsoil4"){
  values(ras)[which(values(ras) < 0)] <- 0}
# SAVING DATA----
setwd(Dir.ERA.Monthly)
writeRaster(ras, paste(Variable, "_mean_", RegionFile, "_", FromM, FromY, "_", ToM, ToY, sep=""),
  overwrite=TRUE, format="CDF", varname=Variable,
  longname= paste(Variables_vec[VarPos], " mean for years ", FromM, "/", FromY, " to ", ToM, "/", ToY,
    " across ", Region, sep=""))
if(Temporary == "Delete"){unlink(Dir.Temp, recursive = TRUE)}
setwd(mainDir)}# end of RasterEra5 function

```

A.3.2.3 Calculation of Vegetation Memory

Chunk 10: Computing vegetation memory for study regions as specified by lags, time frame, and variables to be considered (VegMem), scaling rasters of vegetation memory to be on the same scale for each study region (CoeffScaling).

```

Variables_vec <- c("Tair_mean", "Qsoil1_mean", "Qsoil2_mean", "Qsoil3_mean", "Qsoil4_mean")
VariablesNames_vec <- c("Air Temperature", "Soil Moisture (0-7cm)", "Soil Moisture (7-28cm)",
                       "Soil Moisture (28-100cm)", "Soil Moisture (100-200cm)")

#####----- VegMem [ClimVar, Region, Cumlags, FromY, ToY]
# (selecting data, calculating vegetation memory according to specified lags, exporting rasters) ----
VegMem <- function(ClimVar, ClimVar2, Region, Cumlags, FromY, ToY){
  print("#####
  print(paste("Identifying vegetation memory effects of NDVI based on antecedent NDVI and ",
               VariablesNames_vec[which(Variables_vec == ClimVar2)], " (immediate effects) and ",
               VariablesNames_vec[which(Variables_vec == ClimVar)], " at lags: ", toString(Cumlags), " across ",
               Region, sep=""))

  # LOAD DATA----
  ## NDVI/GIMMs
  NDVI_ras <- brick(paste(Dir.Gimms.Monthly, "/NDVI_", Region, ".nc", sep=""))
  ## ERA5
  Clim <- list.files(Dir.ERA.Monthly)[grep(pattern = ClimVar, list.files(Dir.ERA.Monthly))]
  Clim <- Clim[grep(pattern = Region, Clim)] # files in target region
  Clim <- Clim[grep(pattern = FromY, Clim)] # files with correct start date
  Clim <- Clim[grep(pattern = ToY, Clim)] # files with correct end date
  Clim_mean_ras <- brick(paste(Dir.ERA.Monthly, "/", Clim, sep="")) # rasterise
  Clim2 <- list.files(Dir.ERA.Monthly)[grep(pattern = ClimVar2, list.files(Dir.ERA.Monthly))]
  Clim2 <- Clim2[grep(pattern = Region, Clim2)]
  Clim2 <- Clim2[grep(pattern = FromY, Clim2)]
  Clim2 <- Clim2[grep(pattern = ToY, Clim2)]
  Clim2_mean_ras <- brick(paste(Dir.ERA.Monthly, "/", Clim2, sep=""))

  # PREPARE DATA----
  ## Limit NDVI data to ERA5 time frame
  NDVIYears <- rep(1982:2015, each = 12) # Year vector to indicate months for time frame selection
  NDVITo <- max(which(NDVIYears %in% ToY))
  if(min(which(NDVIYears %in% FromY)) == Inf){
    NDVIFrom <- 1 }else{ NDVIFrom <- min(which(NDVIYears %in% FromY))}

  NDVI_ras <- NDVI_ras[[NDVIFrom:NDVITo]] # NDVI data is limited

  ## Identify data positions
  # establish a mean raster (this sets every cell to NA where any NA is within the time series)
  NATest_ras <- mean(NDVI_ras)
  NATest_vec <- values(NATest_ras) # set values as vector
  Data_Pos <- which(!is.na(NATest_vec)) # select non-NA positions (these are the ones we should build models on)

  # PREPARE RASTERS----
  ModelEval_ras <- NDVI_ras[[1:10]] # select six raster layers
  # put names on the layers to tell us what they contain later
  ModelEval_ras <- Fun_NamesRas(raster = ModelEval_ras, ClimVar = ClimVar, ClimVar2 = ClimVar2)

  # MODELS----
  for(pixel in Data_Pos){ # loop non-NA pixels
    T_Begin <- Sys.time() # note time when calculation is started (needed for estimation of remaining time)
    ## DATA ----
    ### NDVI stuff -----
    NDVI_vecraw <- as.vector(NDVI_ras[pixel]) # extract data
    NDVI_vecdet <- detrend(NDVI_vecraw, tt = 'linear') # linear detrending
    # create NDVI data frame
    NDVI_df <- data.frame(Month = rep(1:12, length(NDVI_vecraw)/12), NDVI_raw = NDVI_vecraw, NDVI_de = NDVI_vecdet)
    ## calculate anomalies (Z-scores) and monthly means
    NDVI_df <- transform(NDVI_df, NDVI_Anomalies = ave(NDVI_de, Month, FUN=scale),
                         NDVI_Threshold = ave(NDVI_raw, Month, FUN=function(t) mean(t, na.rm=TRUE)))
    NDVI_df <- NDVI_df[nrow(NDVI_df):1,] # reverse order to read "present to past"
  }
}

```

```

NDVI_anom <- c(NDVI_df$NDVI_Anomalies, rep(NA, max(Cumlags))) # extract anomalies, adding cumlag NAs
ThreshPos <- which(NDVI_df$NDVI_Threshold < 0.1) # positions which should be excluded
if(length(ThreshPos) == length(NDVI_vecraw)){ # if all months should be masked due to NDVImean < 0.1
  # set all in model raster layers to NA for this pixel
  ModelEval_ras[pixel] <- as.numeric(rep(NA, dim(ModelEval_ras)[3]))
  next()
}

# calculate lag 1
NDVI_Lag1 <- c(NDVI_anom[-1], NA) # adding one NA for month preceeding data range of NDVI itself
#### Climate stuff -----
##### ClimVar -----
Clim_vec <- as.vector(Clim_mean_ras[pixel]) # extract raw data for pixel (instantenous predictor)
Clim_vec <- detrend(Clim_vec, tt = 'linear') # linear detrending
Clim_vec <- Clim_vec[nrow(Clim_vec):1,] # reverse order to read "present to past"
# calculate cumulative climate indices (antecedent predictor)
Clim_cum <- rep(NA, length(Cumlags))
Clim_cum <- as.list(Cumlags)
position <- 1
for(lag in Cumlags){
  for(i in 1:(length(Clim_vec)-lag)){
    Clim_cum[[position]] <- c(Clim_cum[[position]], sum(Clim_vec[i:(i+lag)])))
  }
  Clim_cum[[position]] <- Clim_cum[[position]][-1] # removing initial NA
  # adding enough NAs to bring it up to full length
  Clim_cum[[position]] <- c(Clim_cum[[position]], rep(NA , length(Clim_vec)-length(Clim_cum[[position]]))))
  position <- position+1
}
# make data frame of climate stuff
Clim_df <- as.data.frame(Clim_cum) # make list into data frame
Clim_df <- cbind(rep(12:1, length(Clim_vec)/12), Clim_vec, Clim_df) # append month index and raw data
colnames(Clim_df) <- c("Month", "Clim_raw", paste(rep("ClimCum_",length(Cumlags)),Cumlags, sep="")) # column names
# calculate anomalies
for(anomaly in 2:length(Clim_df)){# cycle through all the columns of the climate data frame except the month column
  Clim_iter <- with(Clim_df, cbind(Month, Clim_df[,anomaly])) # extract necessary data
  colnames(Clim_iter) <- c("Month", "AnomalyCalc") # set column names
  Clim_iter <- transform(Clim_iter, # calculate anomaly for each month
    AnomalyCalc = ave(AnomalyCalc, Month, FUN=scale))
  # save to original data frame
  Clim_df[,anomaly] <- Clim_iter$AnomalyCalc}
##### ClimVar2 -----
Clim2_vec <- as.vector(Clim2_mean_ras[pixel]) # extract raw data for pixel (instantenous predictor)
Clim2_vec <- detrend(Clim2_vec, tt = 'linear') # linear detrending
Clim2_df <- data.frame(Month = rep(1:12, length(Clim2_vec)/12), Clim2_raw = Clim2_vec)
# calculate anomalies
Clim2_df <- transform(Clim2_df,
  Clim2_Anomalies = ave(Clim2_raw, Month, FUN=scale))
Clim2_df <- Clim2_df[nrow(Clim2_df):1,] # reverse order to read "present to past"
Clim2_anom <- Clim2_df$Clim2_Anomalies
### Combining all the data -----
ModData_df <- cbind(NDVI_anom[1:length(Clim2_anom)], NDVI_Lag1[1:length(Clim2_anom)], Clim_df, Clim2_anom)
if(length(ThreshPos) > 0){ # set threshold months to NA if necessary
  ModData_df$NDVI_anom[ThreshPos] <- NA}
ModData_df <- na.omit(ModData_df) # get rid of NA rows
## MODELS -----
### Establishing models-----
# list to save Model objects
Mods_ls <- as.list(rep(NA, length(Cumlags))) # List of models
ps <- rep(NA, length(Cumlags)) # p-values
coeffst1 <- rep(NA, length(Cumlags)) # coefficients of NDVI-1
coeffsC <- rep(NA, length(Cumlags)) # coefficients of ClimVar
coeffsC2 <- rep(NA, length(Cumlags)) # coefficients of ClimVar2

```

```

# iterate over all climate lags
counter <- 0 # create a counter variable
for(ModelIter in Cumlags){ # go through all possible cumulative lags
  ## PCA of our variables
  pca_mat <- matrix(cbind(ModData_df$NDVI_Lag1,ModData_df[, counter+4], ModData_df$Clim2_anom),
                      ncol = 3, byrow = FALSE, dimnames = list(1:length(ModData_df$NDVI_Lag1),
                                                               c("t-1", ClimVar, ClimVar2))) # pca matrix

  pca <- rda(pca_mat) # running pca
  ## Extracting PC axes
  pc1 <- summary(pca)[["sites"]][,1]
  pc2 <- summary(pca)[["sites"]][,2]
  pc3 <- summary(pca)[["sites"]][,3]
  ## Building models
  Mod0 <- lm(ModData_df$NDVI_anom ~ 1) # null model
  Mod <- lm(ModData_df$NDVI_anom ~ pc1 + pc2 + pc3) # full model
  loadings <- summary(pca)[["species"]] # extract loadings
  coefficients <- Mod$coefficients[2:(dim(summary(pca)[["sites"]]) [2]+1)] # extract coefficients
  ## Make coefficients representative by multiplying them with the loadings
  tnewCof <- loadings[,1] * coefficients
  CnewCof <- loadings[,2] * coefficients
  C2newCof <- loadings[,3] * coefficients
  ## Saving information to vectors
  coeffst1[counter+1] <- sum(tnewCof) # NDVI-1
  coeffsC[counter + 1] <- sum(CnewCof) # ClimVar
  coeffsC2[counter + 1] <- sum(C2newCof) # ClimVar2
  if(anova(Mod0, Mod)$RSS[1] > anova(Mod0, Mod)$RSS[2]){ # only save p value if model is an improvement
    ps[counter+1] <- anova(Mod0, Mod)$'Pr(>F)'[2]
  }else{ # if model is not an improvement over null, set p to 1
    ps[counter+1] <- 1
  }
  Mods_ls[[counter+1]] <- Mod # save model to list of models
  counter <- counter + 1 }

  #### Selecting best model -----
  AICs <- sapply(X = Mods_ls, FUN = AIC) # calculate AICs for each model
  Best <- which(abs(AICs) == min(abs(AICs), na.rm = TRUE))[1] # best model, if same value present use first
  c_NDVI <- coeffst1[Best] # ndvi coefficient
  c_Clim <- coeffsC[Best] # climate coefficient
  c_Clim2 <- coeffsC2[Best] # p-value of climate coefficient
  p_Mod <- ps[Best] # p-value is set to p-value of best model
  AICMod <- AICs[Best] # AIC is set to best AIC
  Bestlag <- Cumlags[Best] # this is the lag at which best model was observed
  ## EXPLAINED VARIANCE-----
  colnames(ModData_df)[c(1:2)] <- c("NDVI_anom", "NDVI_Lag1")
  # Legendre & Legendre
  Explainedvar <- lm(data = ModData_df,
                       NDVI_anom ~ NDVI_Lag1 + ModData_df[,Best+5])
  Explainedvar <- summary(Explainedvar)[["r.squared"]]
  VarTotalNDVI <- lm(data = ModData_df,
                        NDVI_anom ~ NDVI_Lag1)
  VarTotalNDVI <- summary(VarTotalNDVI)[["r.squared"]]
  VarTotalQsoil <- lm(data = ModData_df,
                        NDVI_anom ~ ModData_df[,Best+5])
  VarTotalQsoil <- summary(VarTotalQsoil)[["r.squared"]]
  VarShared <- VarTotalQsoil + VarTotalNDVI - Explainedvar
  VarNDVI <- VarTotalNDVI - VarShared
  VarQsoil <- VarTotalQsoil - VarShared
  ## WRITING INFORMATION TO RASTERS-----
  ModelEval_ras[pixel] <- as.numeric(c((Bestlag), AICMod, p_Mod, c_NDVI, c_Clim, c_Clim2, Explainedvar,
                                         VarNDVI, VarShared, VarQsoil)) # saving model information to raster

```

```

## Updating progress bar----
if(pixel == Data_Pos[1]) { # if we are currently on the first pixel
  T_End <- Sys.time() # note end time
  Duration <- as.numeric(T_End)-as.numeric(T_Begin) # calculate the time it took to establish and select models
  ## Put an estimator up on the console that tells the user when to expect the program to finish its current run
  print(paste("Calculating Vegetation Memory effects across ", Region, " should finish around: ",
             as.POSIXlt(T_Begin + Duration*length(Data_Pos), tz = Sys.timezone(location=FALSE)), sep=""))
  ## Update progress bar
  pb <- txtProgressBar(min = 0, max = length(Data_Pos), style = 3)
  pbi <- 0
  pbi <- pbi + 1 ## Update progress bar
  setTxtProgressBar(pb, pbi)} # end of pixel loop
#### Save raster ----
writeRaster(ModelEval_ras, filename = paste(Dir.Memory,"/", Region, "_", ClimVar2, "-", ClimVar,
                                              paste(Cumlags, collapse="_"), "_", FromY, "-", ToY, ".nc",sep=""),
            overwrite=TRUE, format="CDF")
setwd(mainDir)}# end of VegMem function

#####----- CoeffScaling [ClimVar, ClimVar2, Region, Cumlags, FromY, ToY, UAbs]
# (loading previously saved rasters and making visualisation of model coefficients better) -----
CoeffScaling <- function(ClimVar, ClimVar2, Region, Cumlags, FromY, ToY, UAbs){
  print("#####-----")
  print(paste("Producing composites of vegetation memory effects across ", unique(Region), sep=""))
  # PREPARATIONS ----
  Rasters <- ClimVar
  minmaxNDVIn <- rep(NA, length(ClimVar)*2)
  minmaxNDVIIs <- rep(NA, length(ClimVar)*2)
  minmaxCVn <- rep(NA, length(ClimVar)*2)
  minmaxCVs <- rep(NA, length(ClimVar)*2)
  minmaxCV2n <- rep(NA, length(ClimVar)*2)
  minmaxCV2s <- rep(NA, length(ClimVar)*2)
  minmaxPos <- 1
  # LOADING DATA ----
  for(rasiter in 1:length(ClimVar)){ # cycle through specified vegetation memory rasters
    # load raster
    Alter_ras <- brick(paste(Dir.Memory,"/", Region[[rasiter]], "_", ClimVar2[[rasiter]], "-", ClimVar[[rasiter]],
                               paste(Cumlags[[rasiter]], collapse="_"), "_", FromY, "-", ToY, ".nc",sep=""))
    Alter_ras <- Fun_NamesRas(raster = Alter_ras, ClimVar = ClimVar, ClimVar2 = ClimVar2, rasiter = rasiter)
    # PREPARING DATA ----
    P_ras <- Alter_ras$Model.p.value # extract p-value layer
    C_clim <- Alter_ras[[5]] # extract ClimVar coefficients
    C_climNon <- C_clim
    C_climNon[which(values(P_ras) < 0.05)] <- NA # set everything that significant to NA
    C_climSig <- C_clim
    C_climSig[which(values(P_ras) >= 0.05)] <- NA # set everything that's not significant to NA
    C_clim2 <- Alter_ras[[6]] # extract ClimVar2 coefficients
    C_climNon2 <- C_clim2
    C_climNon2[which(values(P_ras) < 0.05)] <- NA # set everything that significant to NA
    C_climSig2 <- C_clim2
    C_climSig2[which(values(P_ras) >= 0.05)] <- NA # set everything that's not significant to NA
    C_NDVI <- Alter_ras$Antecedent.NDVI..c_NDVI. # extract NDVI-1 coefficients
    C_NDVINon <- C_NDVI
    C_NDVINon[which(values(P_ras) < 0.05)] <- NA # set everything that significant to NA
    C_NDVISig <- C_NDVI
    C_NDVISig[which(values(P_ras) >= 0.05)] <- NA # set everything that's not significant to NA
    C_Lags <- Alter_ras[[1]] # extract Lags coefficients
    C_LagsNon <- C_Lags
    C_LagsNon[which(values(P_ras) < 0.05)] <- NA # set everything that significant to NA
  }
}

```

```

C_LagsSig <- C_Lags
C_LagsSig[which(values(P_ras) >= 0.05)] <- NA # set everything that's not significant to NA
# SAVING PARAMETERS ----
Rasters[[rasiter]] <- list(C_NDVINon, C_NDVISig, C_climNon, C_climSig,
                           C_climNon2, C_climSig2, C_LagsNon, C_LagsSig)

## Identify maximum values of each coefficient raster
minmaxNDVIn[minmaxPos] <- max(values(C_NDVINon), na.rm = TRUE)
minmaxNDVIs[minmaxPos] <- max(values(C_NDVISig), na.rm = TRUE)
minmaxCVn[minmaxPos] <- max(values(C_climNon), na.rm = TRUE)
minmaxCVs[minmaxPos] <- max(values(C_climSig), na.rm = TRUE)
minmaxCV2n[minmaxPos] <- max(values(C_climNon2), na.rm = TRUE)
minmaxCV2s[minmaxPos] <- max(values(C_climSig2), na.rm = TRUE)
minmaxPos <- minmaxPos + 1 # +1 to counter in vector

## Identify minimum values of each coefficient raster
minmaxNDVIn[minmaxPos] <- min(values(C_NDVINon), na.rm = TRUE)
minmaxNDVIs[minmaxPos] <- min(values(C_NDVISig), na.rm = TRUE)
minmaxCVn[minmaxPos] <- min(values(C_climNon), na.rm = TRUE)
minmaxCVs[minmaxPos] <- min(values(C_climSig), na.rm = TRUE)
minmaxCV2n[minmaxPos] <- min(values(C_climNon2), na.rm = TRUE)
minmaxCV2s[minmaxPos] <- min(values(C_climSig2), na.rm = TRUE)
minmaxPos <- minmaxPos + 1 # +1 to counter in vector

} # end of loop cycling through specified vegetation memory rasters

# MANN-WHITNEY U ----
## setting up directory
Dir.Memory.Reg <- paste(Dir.Memory,"/",unique(Region),"~",FromY,"_",ToY, sep="")
dir.create(Dir.Memory.Reg)

## cleaning directory o potential earlier runs
if(paste("U-Variables_Abs",UAbs,".xlsx",sep="") %in% list.files(Dir.Memory.Reg)){
  file.remove(paste(Dir.Memory.Reg,"/U-Variables_Abs",UAbs,".xlsx",sep=""))

# Establish matrices and vectors for naming
UModMat <- matrix(rep(NA, length(Rasters)^2), nrow=length(Rasters)) # for saving U outputs
UVariables <- c("NDVI t-1", "Qsoil", "Tair", "Lags") # for naming purposes
UMedians <- matrix(rep(NA, length(Rasters)^2), nrow=length(Rasters)) # for saving variable value medians
dimnames(UMedians) <- list(c(1:4), UVariables) # set names

# variable-wise comparison
for(UVar in 1:length(Rasters)){ # loop over all variables
  for(UTest in 1:(length(Rasters)-1)){ # loop over the model layers
    UTest2 <- UTest + 1 # create separate counter for variable with which to compare
    while(UTest2 <= length(Rasters)){ # cycle so long as second counter does not exceed range of specified models
      if(UAbs == TRUE){ # if absolute values should be used
        Test1 <- abs(values(Rasters[[UTest]][[UVar * 2]])) # data extraction
        Test2 <- abs(values(Rasters[[UTest2]][[UVar * 2]])) # data extraction
      }else{ # if absolute values are not desired
        Test1 <- values(Rasters[[UTest]][[UVar * 2]])
        Test2 <- values(Rasters[[UTest2]][[UVar * 2]])
      }
      test <- wilcox.test(Test1, Test2, paired = FALSE) # WHitney-U Test
      UModMat[UTest, UTest2] <- test$statistic # Extract test statistic
      UModMat[UTest2, UTest] <- test$p.value # Extract p-value
      Med1 <- median(Test1, na.rm=TRUE) # extract median
      Med2 <- median(Test2, na.rm=TRUE) # extract median
      UMedians[UTest,UVar] <- Med1 # write median of first object
      if(UTest2 == length(Rasters)){ # only write median of last variable
        UMedians[UTest2,UVar] <- Med2}# if statement
      UTest2 <- UTest2 + 1 } # while statement
    } # for UTest statement
    # saving output
    write.xlsx(UModMat, sheetName = UVariables[UVar],
               file = paste(Dir.Memory.Reg,"/U-Variables_Abs",UAbs,".xlsx",sep=""), append = TRUE)
  }
}

```

```

} # for UVar statement
write.xlsx(UMedians, sheetName = "Variable medians", # saving output
           file = paste(Dir.Memory.Reg,"/U-Variables_Abs",UAbs,".xlsx",sep=""), append = TRUE)
UModelMat <- matrix(rep(NA,3^2),nrow=3) # matrix for model-internal comparisons
dimnames(UModelMat) <- list(UVariables[1:3], UVariables[1:3]) # set names
# model-wise comparison
for(UTest in 1:(length(Rasters))){ 
  if(UAbs == TRUE){ # if absolute values should be used
    ND <- abs(values(Rasters[[UTest]][[2]]))
    QS <- abs(values(Rasters[[UTest]][[4]]))
    TA <- abs(values(Rasters[[UTest]][[6]]))
  }else{ # if absolute values are not desired
    ND <- values(Rasters[[UTest]][[2]])
    QS <- values(Rasters[[UTest]][[4]])
    TA <- values(Rasters[[UTest]][[6]])}
  UModelMat[2,1] <- wilcox.test(ND, QS, paired = FALSE)$p.value # WHitney-U Test
  UModelMat[3,2] <- wilcox.test(QS, TA, paired = FALSE)$p.value # WHitney-U Test
  UModelMat[3,1] <- wilcox.test(ND, TA, paired = FALSE)$p.value # WHitney-U Test
  UModelMat[1,2] <- wilcox.test(ND, QS, paired = FALSE)$statistic # WHitney-U Test
  UModelMat[2,3] <- wilcox.test(QS, TA, paired = FALSE)$statistic # WHitney-U Test
  UModelMat[1,3] <- wilcox.test(ND, TA, paired = FALSE)$statistic # WHitney-U Test
  write.xlsx(UModelMat, sheetName = paste("Model", UTest, sep=" "), # saving output
             file = paste(Dir.Memory.Reg,"/U-Variables_Abs",UAbs,".xlsx",sep=""), append = TRUE)
} # for UTest statement
# SAVING DATA FOR LATER PLOTTING ----
for(iterplot in 1:length(ClimVar)){ # cycle through all specified vegetation memory raster for plotting
  ## lags -----
  Lag_ras <- brick(paste(Dir.Memory,"/", Region[[iterplot]], "_", ClimVar2[[iterplot]], "-", ClimVar[[iterplot]],
                           paste(Cumlags[[iterplot]], collapse="_"), "_", FromY, "-", ToY, ".nc",sep=""))
  ## NDVI[t-1] -----
  Rasters[[iterplot]][[1]][1] <- max(minmaxNDVIn)
  Rasters[[iterplot]][[1]][2] <- min(minmaxNDVIn)
  Rasters[[iterplot]][[2]][3] <- max(minmaxNDVIs)
  Rasters[[iterplot]][[2]][4] <- min(minmaxNDVIs)
  ### ClimVar -----
  Rasters[[iterplot]][[3]][1] <- max(minmaxCVn)
  Rasters[[iterplot]][[3]][2] <- min(minmaxCVn)
  Rasters[[iterplot]][[4]][3] <- max(minmaxCVs)
  Rasters[[iterplot]][[4]][4] <- min(minmaxCVs)
  ### ClimVar2 -----
  Rasters[[iterplot]][[5]][1] <- max(minmaxCV2n)
  Rasters[[iterplot]][[5]][2] <- min(minmaxCV2n)
  Rasters[[iterplot]][[6]][3] <- max(minmaxCV2s)
  Rasters[[iterplot]][[6]][4] <- min(minmaxCV2s)
  ### significant coefficient rasters (already with max/min dots)
  Save_ras <- brick(Lag_ras[[1]], # Lags
                     Rasters[[iterplot]][[2]], # NDVI
                     Rasters[[iterplot]][[4]], # climvar
                     Rasters[[iterplot]][[6]]) # climvar2
  ### Saving significant effects -----
  writeRaster(Save_ras, filename = paste(Dir.Memory.Reg ,"/", ClimVar[[iterplot]], "_", ClimVar2[[iterplot]],
                                         paste(Cumlags[[iterplot]], collapse="_"), "Plots.nc",sep=""),
              overwrite=TRUE, format="CDF")} # plotting loop
setwd(mainDir)} # CoeffScaling end

```

A.3.3 Plant Functional Traits

Chunk 11: Calculating species-specific PFT means (PFTs), creating distribution maps from floral occurrence data obtained from GBIF (DistMaps), and combine occurrence maps with species trait means (PFTRasters).

```
#####----- PFTs []
# (loading data, building species-specific trait means and saving the result) ----
PFTs <- function(){
  # LOADING DATA ---
  ## NDVI (reference raster) ---
  NDVI_ras <- brick(paste(Dir.Gimms.Monthly, "/GlobalNDVI_20112015.nc", sep=""))
  ref_ras <- NDVI_ras[[6]]
  ref_ras[which(values(ref_ras) > -1)] <- 8888 # identify land pixels
  ## Master PFT data from TRY
  PFT_Master <- read.table(file = paste(Dir.TRY, "/4704.txt", sep=""), stringsAsFactors = FALSE, fill = TRUE,
                            sep="\t", header = TRUE)
  ## Extracting necessary data to handle smaller data frame
  PFTs_df <- data.frame(Species = PFT_Master$SpeciesName, ObsID = PFT_Master$ObservationID,
                        Variable = PFT_Master$DataName, Value = PFT_Master$OrigValueStr, Unit = PFT_Master$UnitName)
  ## fixing factor to numeric
  as.numeric.factor <- function(x) {as.numeric(levels(x))[x]}
  PFTs_df$Value <- as.numeric.factor(PFTs_df$Value)
  # CALCULATING SPECIES-SPECIFIC MEAN TRAIT VALUES ---
  ## Preparations for calculations
  Species <- unique(PFTs_df$Species) # all species to consider
  FullSpec_df <- data.frame(Species = NA, Nmass = NA, Height = NA)
  ## Calculations
  for(Iter in 1:length(Species)){ # loop over all species that need consideration
    T_Begin <- Sys.time() # read start time (needed for expected finishing time)
    ### Nitrogen mean
    Nitro <- which(PFTs_df$Species == Species[Iter] & # positions of nitrogen rows for species
                    PFTs_df$Variable == "Leaf nitrogen content per dry mass (Nmass)")
    Mean_Nitro <- mean(PFTs_df$Value[Nitro], na.rm = TRUE) # mean
    ### Height mean
    Height <- which(PFTs_df$Species == Species[Iter] & # positions of height rows for species
                     PFTs_df$Variable == "Plant height vegetative")
    Mean_Height <- mean(PFTs_df$Value[Height], na.rm = TRUE) # mean
    ### Combining data into a data frame
    Spec_df <- data.frame(Species = Species[Iter], Nmass = Mean_Nitro, Height = Mean_Height)
    FullSpec_df <- rbind(FullSpec_df, Spec_df)
    ### Updating progress bar
    if(Iter == 1){ # estimate on finishing time on first loop
      T_End <- Sys.time() # read end time
      Duration <- as.numeric(T_End)-as.numeric(T_Begin) # duration between date points
      ### Output to console
      print(paste("Calculating species-specific trait mean values should finish around ",
                  as.POSIXlt(T_Begin + Duration*length(Species), tz = Sys.timezone(location=TRUE)), sep=""))
      ### Removing empty initial row
      FullSpec_df <- FullSpec_df[-1,]# end of estimator if-statement
      pb <- txtProgressBar(min = 0, max = length(Species), style = 3) # Setting up a progress bar
      setTxtProgressBar(pb, Iter)}# end of species-specific mean trait value calculation for-loop
    FullNA <- which(is.na(FullSpec_df$Nmass) & is.na(FullSpec_df$Height)) # identifying species where both means are NA
    CorrectedSpec_df <- FullSpec_df[-FullNA,] # removing full NA species records
    # Save the species-specific and NA-free data
    save(CorrectedSpec_df, file = paste(Dir.TRY, "/SpeciesTraits.RData", sep=""))
    setwd(mainDir)

  # BUILDING RASTERS FROM RAW TRAIT MEASURES ---
  ## create empty data frame and filling it
```

```

Locs_df <- data.frame(H = NA, Nmass = NA, Lat = NA, Lon = NA)
`%nin%` = Negate(`%in%`) # create a 'not in' statement
## eliminate observations with less than three records (these can't have enough data)
IDs <- PFTs_df$ObsID
exclude <- names(table(IDs))[which(table(IDs) < 3)]
IDs <- unique(IDs)[which(as.character(unique(IDs)) %nin% exclude)]
## loop over all observations with enough data
for(i in 1:length(IDs)){
  T_Begin <- Sys.time() # read start time (needed for expected finishing time)
  Iter_df <- PFTs_df[which(PFTs_df$ObsID == IDs[i],)] # extract all data for current observation
  ## data extraction
  Nmass <- Iter_df$Value[which(Iter_df$Variable == "Leaf nitrogen content per dry mass (Nmass)")]
  if(length(Nmass) == 0){Nmass <- NA}
  H <- Iter_df$Value[which(Iter_df$Variable == "Plant height vegetative")]
  if(length(H) == 0){H <- NA}
  Lat <- as.numeric(Iter_df$Value[which(Iter_df$Variable == "Latitude")])
  Lon <- as.numeric(Iter_df$Value[which(Iter_df$Variable == "Longitude")])
  Locs_df <- rbind(Locs_df, c(H, Nmass, Lat, Lon)) # bin data
  ## Updating progress bar
  if(i == 1){ # estimate on finishing time on first loop
    T_End <- Sys.time() # read end time
    Duration <- as.numeric(T_End)-as.numeric(T_Begin) # duration between date points
    ### Output to console
    print(paste("Extracting raw geo-referenced data should finish around ",
               as.POSIXlt(T_Begin + Duration*length(IDs), tz = Sys.timezone(location=TRUE)), sep=""))
    ### Removing empty initial row
    Locs_df <- Locs_df[-1,]
    pb <- txtProgressBar(min = 0, max = length(IDs), style = 3)
    setTxtProgressBar(pb, i)
  } # obuservation loop
  ## converting to spatial points objects
  H_df <- na.omit(Locs_df[which(!is.na(Locs_df$H)), -2])
  H_pts <- data.frame(y = H_df$Lat, x = H_df$Lon, z = H_df$H)
  H_pts <- na.omit(H_pts)
  coordinates(H_pts) = ~x+y # convert x and y to coordinates
  Nmass_df <- na.omit(Locs_df[which(!is.na(Locs_df$Nmass)), -1])
  Nmass_pts <- data.frame(y = Nmass_df$Lat, x = Nmass_df$Lon, z = Nmass_df$Nmass)
  Nmass_pts <- na.omit(Nmass_pts)
  coordinates(Nmass_pts) = ~x+y # convert x and y to coordinates
  ## rasterising
  rast <- raster(ext=extent(ref_ras), resolution=res(ref_ras)) # create raster to be filled
  H_rasOut <- rasterize(x = H_pts, y = rast, field = H_pts$z, fun = mean) # rasterize irregular points
  Nmass_rasOut <- rasterize(x = Nmass_pts, y = rast, field = Nmass_pts$z, fun = mean) # rasterize irregular points
  Means_ras <- brick(H_rasOut, Nmass_rasOut)
  ## saving data
  writeRaster(x=Means_ras, filename = paste(Dir.TRY, "/RawTRY-Global", sep=""), overwrite=TRUE, format="CDF")
}# end of PFTs-function

#####----- DistMaps [Species, Extent, Years, CountCodes]
# (Obtaining occurrence data via GBIF, rasterising, saving the raster, limitting to a region) ----
DistMaps <- function(Species, Years, CountCodes){
  print("#####-----")
  print(paste("Downloading occurence data of species: ", Species, " across: ", CountCodes, sep=""))
  # LOADING SPECIES DATA FRAME ----
  load(paste(Dir.TRY, "/SpeciesTraits.RData", sep="")) # load data frame 'CorrectedSpec_df'
  if(Species == "All"){ # selecting all species contained in 'CorrectedSpec_df'
    Species <- sort(CorrectedSpec_df$Species)
    Species <- Species[-which(Species == "-")] # remove this error of a species name
  }
}

```

```

SP <- "All"
}else{
  SP <- "Dummy" # only used if single species is targeted
}
# GLOBAL REFERENCE DATA (needed for rasterising and masking) ----
## NDVI (reference raster) ---
NDVI_ras <- brick(paste(Dir.Gimms.Monthly, "/GlobalNDVI_20112015.nc", sep=""))
ref_ras <- NDVI_ras[[6]]
ref_ras[which(values(ref_ras) > -1)] <- 8888 # identify land pixels
# IDENTIFYING GBIF KEY(S) ----
# if data is already present and all species are sought-after
if("SpeciesGBIFKeys.rda" %in% list.files(path=Dir.TRY) & SP == "All"){
  print("Loading species-specific GBIF keys from local storage") # output to console
  load(paste(Dir.TRY, "/SpeciesGBIFKeys.rda", sep=""))
}else{
  ## Preparations ---
  Key_vec <- NA # create empty vector for gbif key(s)
  Species_Pres <- NA # create empty vector for all species which we have occurrence records for
  print("Identifying species-specific GBIF keys from GBIF repository") # output to console
  pb <- txtProgressBar(min = 0, max = length(Species), style = 3) # Setting up a progress bar
  for(Iter in 1:length(Species)){ # cycle through all species specified
    key <- name_suggest(q=Species[Iter], rank='species')$key[1] # pull gbif key
    Key_vec <- c(Key_vec, key) # append key to key vector
    if(!is.null(key)){ # if we have occurrence data in the gbif records
      Species_Pres <- c(Species_Pres, Species[Iter]) # append species to species vector
    }
    setTxtProgressBar(pb, Iter)
  }
  ## Fixing vectors
  Key_vec <- Key_vec[-1] # Removing empty initial element
  Species_Pres <- Species_Pres[-1] # removing empty initial element
  # Save the species-specific and NA-freed data
  if(SP == "All"){ # if we want all species, we might as well save the names and key objects for later saving of time
    save(list = c("Species_Pres", "Key_vec"), file = paste(Dir.TRY, "/SpeciesGBIFKeys.rda", sep=""))} # GBIF keys
# OCCURENCE DATA ----
## Preparation
print("Downloading species-specific occurrence records from GBIF") # output to console
pb <- txtProgressBar(min = 0, max = length(Species_Pres), style = 3) # Setting up a progress bar
## If an error occurred previously
if("Breakage.txt" %in% list.files(path = mainDir)){ # this file is only present if the run finished prematurely
  OccIter <- read.table(paste(mainDir, "/Breakage.txt", sep=""))[1,1] # position at which it failed previously - 1
}else{
  OccIter <- 0 # set to 0 if it didn't fail previously
}
if(OccIter > 1){ # if previous run (OccIter) failed at the second step or later
  Start <- OccIter + 1 # start from where it failed, OccIter is the last one that got done
}else{ # if it failed at the first one
  Start <- 1 # start at the first species
}
for(OccIter in Start:length(Key_vec)){ # cycling through all species to obtain occurrence data
  # if species name cannot be put into a file name due to special characters, this excludes 61 species records
  if(grepl('^[[:alnum:]]+\\.-', Species_Pres[OccIter]) == TRUE){
    next()
  }
  if(paste(Species_Pres[OccIter], "_", CountCodes, ".rda", sep="") %in% list.files(path=Dir.OCCs)){
  }else{ # data not present locally yet
    ## Downloading Data
    key <- Key_vec[OccIter] # select GBIF key
    Gbif <- occ_data(key, limit=200000, hasCoordinate = TRUE, year = Years,
                     hasGeospatialIssue = FALSE, country = CountCodes) # download data
  }
}

```

```

## Dealing with separate data frames of years
BaseOcc <- rep(NA, 3) # create empty vector
BaseOcc_df <- t(as.data.frame(BaseOcc)) # make empty vector into empty data frame
colnames(BaseOcc_df) <- c("decimalLatitude", "decimalLongitude", "year") # set column names
for(i in 1:length(Years)){
  # create a data frame of latitude and longitude records of currently iterated year
  GbifFrame <- data.frame(decimalLatitude = Gbif[[i]]$data$decimalLatitude,
                          decimalLongitude = Gbif[[i]]$data$decimalLongitude,
                          year = rep(Years[i], length(Gbif[[i]]$data$decimalLatitude)))
  BaseOcc_df <- rbind(BaseOcc_df, GbifFrame)}
## Sanity check
if(dim(BaseOcc_df)[1] == 1){ # if there is no occurrence data
  next()}
BaseOcc_df <- BaseOcc_df[-1,] # remove initial NA row
## Saving data frame
save(BaseOcc_df, file = paste(Dir.OCCs, "/", Species_Pres[[OccIter]], "_", CountCodes, ".rda", sep=""))
setTxtProgressBar(pb, OccIter) # update progress bar
# save current iteration number to disk (used for jumping right back in if errors occur)
write.table(OccIter, file = paste(mainDir, "/Breakage.txt", sep=""))
}# occurrence data loop
file.remove(paste(mainDir, "/Breakage.txt", sep=""))
setwd(mainDir)
GbifStat <- "Done"
return(GbifStat)
}# end of Mapping function

#####----- PFTRasters [Region, Extent, RegionFile, CountCodes]
# (loading data, building species-specific trait mean rasters for study regions) -----
PFTRasters <- function(Region, Extent, RegionFile, CountCodes){
  print("#####")
  print(paste("Building mean trait rasters across ", RegionFile, sep=""))
  load(paste(Dir.TRY, "/SpeciesTraits.RData", sep="")) # load data
  RawTry_ras <- brick(paste(Dir.TRY, "/RawTRY-Global.nc", sep=""))
  # GLOBAL REFERENCE DATA (needed for rasterising and masking) -----
  ## NDVI (reference raster) ---
  NDVI_ras <- brick(paste(Dir.Gimms.Monthly, "/GlobalNDVI_20112015.nc", sep=""))
  ref_ras <- NDVI_ras[[6]]
  ref_ras[which(values(ref_ras) > -1)] <- 8888 # identify land pixels
  # REGION SELECTION---
  Shapes <- readOGR(Dir.Mask, 'ne_50m_admin_0_countries', verbose = FALSE)
  RegObj <- RegionSelection(Region = Region, RegionFile = RegionFile, Extent = Extent)
  area <- RegObj[[1]]
  location <- RegObj[[2]]
  RegionFile <- RegObj[[3]]
  # CROPPING AND MASKING -----
  ## Reference cropping and masking
  ref_rasC <- crop(ref_ras, area) # cropping to extent
  ref_rasF <- mask(ref_rasC, Shapes[location,]) # masking via Shapefile
  # RAW TRY DATA -----
  RawTry_rasC <- crop(RawTry_ras, area) # cropping to extent
  RawTry_rasF <- mask(RawTry_rasC, Shapes[location,]) # masking via Shapefile
  writeRaster(x=RawTry_rasF, filename = paste(Dir.TRY, "/RawTRY-", RegionFile, sep=""), overwrite=TRUE, format="CDF")
  # CALCULATING MEAN RASTERS WITH DISTRIBUTION MAPS -----
  # create empty mean raster
  BaseMeans <- ref_rasF
  values(BaseMeans)[!is.na(values(BaseMeans))] <- 0
  # build brick for mean calculations
  BaseMeans <- brick(BaseMeans, BaseMeans, BaseMeans, BaseMeans)
}

```

```

names(BaseMeans) <- c("Height", "NMass", "HCount", "NCount")
# progress bar
pb <- txtProgressBar(min = 0, max = length(list.files(Dir.OCCs)), style = 3) #
# looping over all .rda occurrence files previously downloaded
for(OccRast in 1:length(list.files(Dir.OCCs))){
  # OCCURENCE ----
  load(paste(Dir.OCCs, "/", list.files(Dir.OCCs)[OccRast], sep=""))
  ## Converting to SpatialPoints
  pts <- data.frame(y = BaseOcc_df$decimalLatitude, x = BaseOcc_df$decimalLongitude,
                     z = rep(1, length(BaseOcc_df$decimalLongitude)))
  pts <- na.omit(pts) # remove NA rows
  coordinates(pts) = ~x+y # convert x and y to coordinates
  # RASTERISING ----
  # create raster to be filled
  rast <- raster(ext=extent(ref_ras), resolution=res(ref_ras))
  # rasterize irregular points
  # we use a mean function here to regularly grid the irregular input points
  rasOut<-rasterize(x = pts, y = rast, field = pts$z, fun = max)
  ## Occurrence cropping and masking
  rasC <- crop(rasOut, area) # cropping to extent
  rasF <- mask(rasC, Shapes[location,]) # masking via Shapefile
  # TRAIT MEANS ----
  # loading data of currently iterated on species
  Grep <- list.files(Dir.OCCs)[OccRast]
  Grep <- gsub(x = Grep, pattern = CountCodes, replacement = "")
  Grep <- gsub(x = Grep, pattern = ".rda", replacement = "")

  NMass <- CorrectedSpec_df$Nmass[which(CorrectedSpec_df$Species == Grep)]
  Height <- CorrectedSpec_df$Height[which(CorrectedSpec_df$Species == Grep)]
  # BUILDING MAP
  Identifier <- which(!is.na(values(rasF)))
  if(length(NMass) != 0){
    if(!is.nan(NMass)){ # add current species-NMass to raster layer and bump up count by 1
      values(BaseMeans$NMass)[Identifier] <- values(BaseMeans$NMass)[Identifier] + NMass
      values(BaseMeans$NCount)[Identifier] <- values(BaseMeans$NCount)[Identifier] + 1}
    if(length(Height) != 0{
      if(!is.nan(Height)){ # add current species-Height to raster layer and bump up count by 1
        values(BaseMeans$Height)[Identifier] <- values(BaseMeans$Height)[Identifier] + Height
        values(BaseMeans$HCount)[Identifier] <- values(BaseMeans$HCount)[Identifier] + 1}
      setTxtProgressBar(pb, OccRast) # update progress bar
    } # OccRast-loop
    # CALCULATE MEANS ----
    TestHeight <- BaseMeans$Height/BaseMeans$HCount
    values(TestHeight)[which(values(TestHeight) > quantile(values(TestHeight), .95, na.rm = TRUE))] <- NA
    TestNMass <- BaseMeans$NMass/BaseMeans$NCount
    values(TestNMass)[which(values(TestNMass) > quantile(values(TestNMass), .95, na.rm = TRUE))] <- NA
    Means_ras <- brick(TestHeight, TestNMass)
    # SAVING DATA ----
    writeRaster(x=Means_ras, filename = paste(Dir.TRY,"/TRY-",RegionFile, sep=""),overwrite=TRUE, format="CDF")
  }# PFTrasters
}

```

A.3.4 COMPADRE

Chunk 12: Extracting and rasterising COMPADRE data from COMPADRE data base for each study region (RasterCOMPADRE).

```
#####----- RasterCOMPADRE [Variable, Region, RegionFile, Extent]
# (Selecting specified COMPADRE data, rasterising, saving the raster, limitting to a region) ----
RasterCOMPADRE <- function(Variable, Region, RegionFile, Extent){
  print("#####-----")
  print(paste("Rasterising COMPADRE ", Variable, " across ", RegionFile, sep=""))
  # LOADING DATA ----
  Compadre_df <- read.csv(paste(Dir.Compadre, "/allCOMPADREOutput.csv", sep="")) # load data frame
  NDVI_ras <- brick(paste(Dir.Gimms.Monthly, "/GlobalNDVI_20112015.nc", sep="")) # reference raster
  ref_ras <- NDVI_ras[[6]] # select only one years data
  ref_ras[which(values(ref_ras) > -1)] <- 8888 # select only land pixels and set them to -8888
  # REGION SELECTION----
  Shapes <- readOGR(Dir.Mask, 'ne_50m_admin_0_countries', verbose = FALSE)
  RegObj <- RegionSelection(Region = Region, RegionFile = RegionFile, Extent = Extent)
  area <- RegObj[[1]]
  location <- RegObj[[2]]
  RegionFile <- RegObj[[3]]
  # DATA MANIPULATION ----
  if(Variable == "FastSlow"){ # analysis of PCA axes
    FSVars <- c("GenT", "H", "La", "GrowSSD", "ShriSSD", "RepSSD", "S", "R0", "Lmean")
    VariableCol <- match(FSVars, colnames(Compadre_df))
    FSLoads <- list(c(.87,.53,.7,-.8,.04,-.81,-.25,-.03,.12), # PCA 1 according to Salguero-Gomez, 2017
                    c(.15,.27,.28,-.05,-.79,.32,.65,.7,.26)) # PCA 2 according to Salguero-Gomez, 2017
    pts <- na.omit(data.frame(y = Compadre_df$Lat, x = Compadre_df$Lon, z = Compadre_df[,VariableCol]))
    PCA1_df <- t(t(pts[,-1:-2]) * FSLoads[[1]]) # multiplying by first axis loadings
    PCA1_df <- rowSums(PCA1_df) # build sums for single index along PCA 1
    PCA1_df <- cbind(pts[,1:2], PCA1_df) # binding with coordinates
    coordinates(PCA1_df) = ~x+y # convert x and y to coordinates
    PCA2_df <- t(t(pts[,-1:-2]) * FSLoads[[2]]) # multiplying by second axis loadings
    PCA2_df <- rowSums(PCA2_df) # build sums for single index along PCA 2
    PCA2_df <- cbind(pts[,1:2], PCA2_df) # binding with coordinates
    coordinates(PCA2_df) = ~x+y # convert x and y to coordinates
    ## Rasterizing ----
    rast <- raster(ext=extent(ref_ras), resolution=res(ref_ras)) # create raster to be filled
    rasOut1 <-rasterize(x = PCA1_df, y = rast, field = PCA1_df$PCA1_df, fun = mean) # rasterize irregular points
    rasOut2 <-rasterize(x = PCA2_df, y = rast, field = PCA2_df$PCA2_df, fun = mean) # rasterize irregular points
    rasOut <- brick(rasOut1, rasOut2)
    names(rasOut) <- c("FS PCA1", "FS PCA2")
  }else{ # single variable desired
    VariableCol <- which(colnames(Compadre_df) == Variable) # figure out the position of the desired Variable
    pts <- na.omit(data.frame(y = Compadre_df$Lat, x = Compadre_df$Lon, z = Compadre_df[,VariableCol]))
    coordinates(pts) = ~x+y # convert x and y to coordinates
    rast <- raster(ext=extent(ref_ras), resolution=res(ref_ras)) # create raster to be filled
    rasOut<-rasterize(x = pts, y = rast, field = pts$z, fun = mean) # rasterize irregular points
    # CROPPING AND MASKING ----
    rasC <- crop(rasOut, area) ## Occurrence cropping and masking
    rasF <- mask(rasC, Shapes[location,]) # masking via Shapefile
    # DATA EXPORT ----
    Dir.Temp.Compadre <- paste(Dir.Compadre, Variable, sep="/")
    dir.create(Dir.Temp.Compadre)
    values(rasF)[which(values(rasF) == Inf)] <- NA # get rid off Inf values (when dealing with Rho)
    invisible(writeRaster(rasF, filename = paste(Dir.Temp.Compadre, "/", Variable, "_", RegionFile, sep=""),
                          overwrite=TRUE, format="CDF")))
  }# end of RasterCOMPADRE
```

A.4 R Data Visualisation Codes

A.4.1 Preamble

Chunk 13: Preamble needed for processing the following chunks within this manuscript.

```
source("S0a_Packages.R") # loading packages
source("S0b_Directories.R") # setting directories
source("S0c_Functions.R") # Loading miscellaneous functions
col.NDVI <- rev(terrain.colors(100))
col.qsoil <- colorRampPalette(c("yellow", "burlywood", "beige", "turquoise", "deepskyblue"))(100)
col.tair <- colorRampPalette(c("blue", "turquoise", "yellow", "orange", "red"))(100)
SR_cols <- list(col.NDVI, col.tair, col.qsoil, col.qsoil, col.qsoil)
SR_Titles <- list("NDVI", "Air Temperature", "Soil Moisture (0-7cm)", "Soil Moisture (7-28cm)",
  "Soil Moisture (28-100cm)", "Soil Moisture (100-255cm)")
setwd(Dir.KrigCov)
Elevation <- raster(list.files()[2], varname = "Elevation")
ElevationF <- raster(list.files()[1], varname = "Elevation")
```

A.4.2 Data Overview

A.4.2.1 NDVI

Chunk 14: Plotting global NDVI mean for the time frame of 1982 - 2015.

```
setwd(Dir.Gimms.Monthly)
NDVIMean_ras <- mean(brick(list.files()[1:7]), na.rm = TRUE)
plot(NDVIMean_ras, colNA = "black", main = "Mean NDVI 1982 - 2015", cex.main = 2,
  legend.width = 1.5, legend.shrink = 1, axis.args = list(cex.axis = 1.5), cex.axis = 1.5)
```

A.4.2.2 ERA5

Chunk 15: Two by two plotting of global soil moisture indices across four different soil layers all on the same scale.

```
Variable <- c("Qsoil1", "Qsoil2", "Qsoil3", "Qsoil4")
colour = col.qsoil
par(mfrow = c(2,2))
setwd(Dir.ERA)
Era_list <- list(NA, NA, NA, NA)
for(i in 1:4){
  Era5ras <- mean(brick(list.files()[grep(pattern = Variable[i], list.files())]), na.rm = TRUE)
  values(Era5ras)[which(is.na(values(Elevation)))] <- NA
  Era_list[[i]] <- Era5ras
}
Era5ras <- brick(Era_list)
Era5ras[1] <- max(Era5ras)$data$max
for(i in 1:4){
  plot(Era5ras[[i]], colNA = "black", main = paste("Mean", SR_Titles[i+2], "1980 - 2015"),
    cex.main = 2, legend.width = 1.5, legend.shrink=1,
    axis.args=list(cex.axis=1.5), cex.axis = 1.5, col = colour, legend = FALSE)
}
```

Chunk 16: Plotting mean air temperature from 1982 - 2015 globally.

```
Variable <- c("Tair")
colour = col.tair
setwd(Dir.ERA)
Era5ras <- mean(brick(list.files()[grep(pattern = Variable, list.files())]), na.rm = TRUE)
values(Era5ras)[which(is.na(values(Elevation)))] <- NA
plot(Era5ras, colNA = "black", main = paste("Mean", SR_Titles[2], "1980 - 2015"),
  cex.main = 2, legend.width = 1.5, legend.shrink=1,
  axis.args=list(cex.axis=1.5), cex.axis = 1.5, col = colour, legend = TRUE)
```

A.4.2.3 HWSD

Chunk 17: Plotting HWSD elevation data.

```
plot(ElevationF, colNA = "black", cex.main = 2, legend.width = 1.5, legend.shrink = 1,
      axis.args = list(cex.axis = 1.5), cex.axis = 1.5, legend = TRUE, col = terrain.colors(100),
      main = "DEM at GIMMs native resolution")
```

Chunk 18: Loading HWSD data and plotting HWSD slope incline data.

```
Covariates_vec <- c("Slopes1", "Slopes2", "Slopes3", "Slopes4", "Slopes5", "Slopes6", "Slopes7", "Slopes8",
                    "Slope_aspect_N", "Slope_aspect_E", "Slope_aspect_S", "Slope_aspect_W", "Slope_aspect_U",
                    "Elevation")

Cov_fine <- list() # create empty list
for(c in 1:length(Covariates_vec)){ # cycle through all covariates and load the data
  Cov_fine[[c]] <- raster(paste(Dir.KrigCov, "/Co-variates_NativeResolution.nc", sep=""),
                           varname = Covariates_vec[c])
}
Cov_fine <- brick(Cov_fine) # make fine covariate data into one big brick

for(i in 1:13){
  values(Cov_fine[[i]]) <- values(Cov_fine[[i]])/1000
}
par(mfrow = c(3,2))
for(i in 9:13){
  plot(Cov_fine[[i]], colNA = "black", cex.main = 2, legend.width = 1.5, legend.shrink=1,
        axis.args=list(cex.axis=1.5), cex.axis = 1.5, legend = TRUE, col = terrain.colors(100),
        main = Covariates_vec[[i]])
}
```

Chunk 19: Plotting HWSD slope aspect data

```
par(mfrow = c(4, 2))
for (i in 1:8) {
  plot(Cov_fine[[i]]/10, colNA = "black", cex.main = 2, legend.width = 1.5, legend.shrink = 1,
        axis.args = list(cex.axis = 1.5), cex.axis = 1.5, legend = TRUE, col = terrain.colors(100),
        main = Covariates_vec[[i]])
}
```

A.4.2.4 TRY

Chunk 20: Plotting global H TRY data with a background plot of land mass.

```
# Data
setwd(Dir.Gimms.Monthly)
Back_ras <- mean(brick(list.files()[1:7]), na.rm = TRUE) # background raster for later plots
values(Back_ras)[which(!is.na(values(Back_ras)))] <- 8888
RawTry_ras <- brick(paste(Dir.TRY, "/RawTRY-Global.nc", sep = ""))
# Height plot
plot(Back_ras, col = "grey", colNA = "black", main = "Mean Vegetative Height", cex.main = 2,
     legend = FALSE)
plot(RawTry_ras[[1]], col = col.tair, add = TRUE, legend.width = 1.5, legend.shrink = 1,
     axis.args = list(cex.axis = 1.5), cex.axis = 1.5)
```

Chunk 21: Plotting global N_{mass} TRY data with a background plot of land mass.

```
# Nmass plot
plot(Back_ras, col = "grey", colNA = "black", main = "Mean Leaf Nitrogen Mass", cex.main = 2,
     legend = FALSE)
plot(RawTry_ras[[2]], col = col.tair, add = TRUE, legend.width = 1.5, legend.shrink = 1,
     axis.args = list(cex.axis = 1.5), cex.axis = 1.5)
```

A.4.2.5 COMPADRE

Chunk 22: Plotting global locations of COMPADRE data availability at GIMMS resolution.

```
# DATA
Compadre_df <- read.csv(paste(Dir.Compadre, "/allCOMPADREOutput.csv", sep = "")) # load data frame
setwd(Dir.Gimms.Monthly)
Back_ras <- mean(brick(list.files()[1:7]), na.rm = TRUE) # background raster for later plots
values(Back_ras)[which(!is.na(values(Back_ras)))] <- 8888
# RASTERISING
pts <- na.omit(data.frame(y = Compadre_df$Lat, x = Compadre_df$Lon, z = rep(1, dim(Compadre_df)[1])))
coordinates(pts) = ~x + y # convert x and y to coordinates
rast <- raster(extent(Back_ras), resolution = res(Back_ras)) # create raster to be filled
rasOut <- rasterize(x = pts, y = rast, field = pts$z, fun = mean) # rasterize irregular points
# PLOTTING
plot(Back_ras, col = "grey", colNA = "black", main = "COMPADRE Data Locations", cex.main = 2,
     legend = FALSE)
plot(rasOut, col = "red", add = TRUE, legend = FALSE, axis.args = list(cex.axis = 1.5),
     cex.axis = 1.5)
```

A.4.2.6 Study Regions

Chunk 23: Function for plotting data overviews in three-by-two plots for study regions.

```
SRData <- function(Region) {
  ##### LOADING DATA ----
  setwd(Dir.Gimms.Monthly)
  SR_NDVI <- mean(brick(list.files()[grep(pattern = Region, list.files())]), na.rm = TRUE)
  setwd(Dir.ERA.Monthly)
  SR_Qsoil1 <- mean(brick(list.files()[grep(pattern = paste("Qsoil1_mean_", Region,
    sep = "")], list.files()), na.rm = TRUE))
  SR_Qsoil2 <- mean(brick(list.files()[grep(pattern = paste("Qsoil2_mean_", Region,
    sep = "")], list.files()), na.rm = TRUE))
  SR_Qsoil3 <- mean(brick(list.files()[grep(pattern = paste("Qsoil3_mean_", Region,
    sep = "")], list.files()), na.rm = TRUE))
  SR_Qsoil4 <- mean(brick(list.files()[grep(pattern = paste("Qsoil4_mean_", Region,
    sep = "")], list.files()), na.rm = TRUE))
  SR_Tair <- mean(brick(list.files()[grep(pattern = paste("Tair_mean_", Region,
    sep = "")], list.files()), na.rm = TRUE))

  ##### FIXING VALUES ----
  QsoilStack <- stack(SR_Qsoil1, SR_Qsoil2, SR_Qsoil3, SR_Qsoil4)
  SR_Qsoil1[1:2] <- c(min(QsoilStack, na.rm = TRUE)@data@min, max(QsoilStack, na.rm = TRUE)@data@max)
  SR_Qsoil2[1:2] <- c(min(QsoilStack, na.rm = TRUE)@data@min, max(QsoilStack, na.rm = TRUE)@data@max)
  SR_Qsoil3[1:2] <- c(min(QsoilStack, na.rm = TRUE)@data@min, max(QsoilStack, na.rm = TRUE)@data@max)
  SR_Qsoil4[1:2] <- c(min(QsoilStack, na.rm = TRUE)@data@min, max(QsoilStack, na.rm = TRUE)@data@max)

  ##### PLOTTING ----
  Plot_Stack <- stack(SR_NDVI, SR_Tair, SR_Qsoil1, SR_Qsoil2, SR_Qsoil3, SR_Qsoil4)
  par(mfrow = c(3, 2), mai = c(1, 0, 0.5, 0))
  for (i in 1:length(SR_cols)) {
    if (i == 1) {
      plot(Plot_Stack[[i]], colNA = "black", main = paste("Mean", SR_Titles[[i]],
        "1982 - 2015"), cex.main = 3, legend.width = 3, legend.shrink = 1,
        axis.args = list(cex.axis = 2.5), cex.axis = 2, col = SR_cols[[i]],
        axes = TRUE, legend = FALSE)
    } else {
      plot(Plot_Stack[[i]], colNA = "black", main = paste("Mean", SR_Titles[[i]],
        "1981 - 2015"), cex.main = 3, legend.width = 3, legend.shrink = 1,
        axis.args = list(cex.axis = 2.5), cex.axis = 2, col = SR_cols[[i]],
        axes = TRUE, legend = FALSE)
    }
    if (i <= 2) {
      plot(Plot_Stack[[i]], legend.only = TRUE, smallplot = c(0, 0.93, 0.12,
        0.16), horizontal = TRUE, axis.args = list(cex.axis = 2.5), col = SR_cols[[i]])
    }
    if (i == 2) {
      par(mai = c(0.5, 0, 0.5, 0))
    }
  }
  return(SR_Qsoil4)
}
```

Chunk 24: Function for plotting TRY data overviews in two-by-two plots for study regions.

```
TRYRegions <- function(Region) {
  ### LOADING DATA ----
  setwd(Dir.Gimms.Monthly)
  Back_ras <- mean(brick(list.files()[1:7]), na.rm = TRUE) # background raster for later plots
  values(Back_ras)[which(!is.na(values(Back_ras)))] <- 8888
  Titles <- c("Height", "Nitrogen Content")
  ### PLOTTING ----
  par(mfrow = c(2, 2), mai = c(1, 0, 0.5, 0))
  setwd(Dir.TRY)
  Plot_ras <- brick(list.files(Dir.TRY)[grep(pattern = Region, list.files(Dir.TRY))][1]) # Raw file
  Plot1_ras <- brick(list.files(Dir.TRY)[grep(pattern = Region, list.files(Dir.TRY))][2]) # Distrib file
  Back_ras1 <- crop(Back_ras, extent(Plot_ras))
  Back_ras1 <- stack(Back_ras1, Back_ras1)
  names(Back_ras1) <- c("Height", "Nitrogen")
  for (i in 1:2) {
    plot(Back_ras1[[i]], colNA = "black", main = Titles[i], cex.main = 3, legend = FALSE,
         col = "grey")
    plot(Plot_ras[[i]], add = TRUE, legend = FALSE, col = col.tair, axes = TRUE)
    plot(Plot_ras[[i]], legend.only = TRUE, smallplot = c(0, 0.95, 0.075, 0.115),
         horizontal = TRUE, axis.args = list(cex.axis = 2.5), col = col.tair)
  }
  for (i in 1:2) {
    plot(Back_ras1[[i]], colNA = "black", main = Titles[i], cex.main = 3, legend = FALSE,
         col = "grey")
    plot(Plot1_ras[[i]], add = TRUE, legend = FALSE, col = col.tair, axes = TRUE)
    plot(Plot1_ras[[i]], legend.only = TRUE, smallplot = c(0, 0.95, 0.075, 0.115),
         horizontal = TRUE, axis.args = list(cex.axis = 2.5), col = col.tair)
  }
}
```

A.4.3 Vegetation Memory Models

Chunk 25: Plotting NDVI data treatment in two-by-two plots.

```
## load data
IbRas <- grep(list.files(Dir.Gimms.Monthly), pattern = "Iberian Region")
IBNDVI <- brick(paste(Dir.Gimms.Monthly, list.files(Dir.Gimms.Monthly)[IbRas],
  sep = "/"))
## extract data
NDVI_vecraw <- as.vector(IBNDVI[35363])
NDVI_vecdet <- detrend(NDVI_vecraw, tt = "linear") # linear detrending
NDVI_df <- data.frame(Month = rep(1:12, length(NDVI_vecraw)/12),
  NDVI_raw = NDVI_vecraw, NDVI_de = NDVI_vecdet) # create NDVI data frame
## calculate anomalies (Z-scores) and monthly means
NDVI_df <- transform(NDVI_df, NDVI_Anomalies = ave(NDVI_de,
  Month, FUN = scale), NDVI_Threshold = ave(NDVI_raw,
  Month, FUN = function(t) mean(t, na.rm = TRUE)))
## plotting data frame
NDVIplot_df <- data.frame(NDVI = c(NDVI_vecraw, NDVI_vecdet,
  NDVI_df$NDVI_Anomalies, NDVI_df$NDVI_Threshold[1:12]),
  Ident = c(rep("Raw Data", length(NDVI_vecraw)),
  rep("Detrended", length(NDVI_vecraw)), rep("Z-Scores",
  length(NDVI_vecraw)), rep("Monthly Means",
  12)), Months = c(rep(1:length(NDVI_vecraw),
  3), 1:12))
## raw and detrended
ggplot(NDVIplot_df[1:(length(NDVI_vecraw) * 2), ],
  aes(x = Months, y = NDVI, col = Ident)) + geom_line(size = 2) +
  theme_bw(base_size = 28) + theme(legend.position = "none") +
  scale_color_manual(values = c("orange", "forestgreen"))
## monthly means
ggplot(NDVIplot_df[(length(NDVI_vecraw) * 3 + 1):(dim(NDVIplot_df)[1]),
  ], aes(x = as.factor(Months), y = NDVI, col = Ident)) +
  geom_bar(stat = "identity", fill = "forestgreen") +
  theme_bw(base_size = 28) + theme(legend.position = "none") +
  scale_color_manual(values = c("forestgreen")) +
  xlab("Months") + geom_hline(aes(yintercept = 0.1),
  size = 1.5)
## z-scores
ggplot(NDVIplot_df[(length(NDVI_vecraw) * 2 + 1):(length(NDVI_vecraw) *
  3), ], aes(x = Months, y = NDVI, col = Ident)) +
  geom_line(size = 2) + theme_bw(base_size = 28) +
  theme(legend.position = "none") + scale_color_manual(values = c("purple"))
## legend
leg <- ggplot(NDVIplot_df[1:(length(NDVI_vecraw) *
  3), ], aes(x = Months, y = NDVI, col = Ident)) +
  geom_line(size = 2) + scale_color_manual(values = c("orange",
  "forestgreen", "purple")) + theme_bw(base_size = 65) +
  guides(colour = guide_legend(override.aes = list(size = 20))) +
  labs(col = "GIMMs NDVI 3g data")
legend <- cowplot::get_legend(leg)
grid.newpage()
grid.draw(legend)
```

Chunk 26: Plotting Qsoil 1 data treatment in two-by-two plots.

```

setwd(Dir.ERA.Monthly)
Qsoil <- brick(list.files()[grep(list.files(), pattern = "Iberian Region_11981_122015.nc")[1]])
load(paste(Dir.Data, "ModData_df.RData", sep = "/")) # model data for pixel 35363
Clim <- as.vector(Qsoil[35363])[1:dim(ModData_df)[1]]
Clim_vec <- detrend(Clim, tt = "linear") # linear detrending
## raw and detrended
plot1_df <- with(ModData_df, data.frame(Months = rep(1:dim(ModData_df)[1],
  2), Qsoil = c(Clim, Clim_vec), Ident = rep(c("Raw Data",
  "Detrened"), each = dim(ModData_df)[1])))
ggplot(plot1_df, aes(x = Months, y = Qsoil, col = Ident)) +
  geom_line(size = 2) + theme_bw(base_size = 28) +
  theme(legend.position = "none") + scale_color_manual(values = c("orange",
  "navyblue"))
## raw and detrended
plot2_df <- with(ModData_df, data.frame(Months = rep(1:dim(ModData_df)[1],
  1), Qsoil = c(Clim_raw), Ident = rep(c("Z-Scores"),
  each = dim(ModData_df)[1])))
ggplot(plot2_df, aes(x = Months, y = Qsoil, col = Ident)) +
  geom_line(size = 2) + theme_bw(base_size = 28) +
  theme(legend.position = "none") + scale_color_manual(values = c("brown"))
## Cummulatve soil moisture lags
plot3_df <- with(ModData_df, data.frame(Months = rep(1:dim(ModData_df)[1],
  13), Qsoil = c(ClimCum_0, ClimCum_1, ClimCum_2,
  ClimCum_3, ClimCum_4, ClimCum_5, ClimCum_6, ClimCum_7,
  ClimCum_8, ClimCum_9, ClimCum_10, ClimCum_11, ClimCum_12),
  Ident = as.factor(rep(c("Z-Scores", "Lag 01", "Lag 02",
  "Lag 03", "Lag 04", "Lag 05", "Lag 06", "Lag 07",
  "Lag 08", "Lag 09", "Lag 10", "Lag 11", "Lag 12"),
  each = dim(ModData_df)[1])))
ggplot(plot3_df, aes(x = Months, y = Qsoil, col = Ident)) +
  geom_line(size = 1) + theme_bw(base_size = 28) +
  theme(legend.position = "none") + scale_color_manual(values = c(rainbow(12),
  "brown"))
## legend
Qsoilplot_df <- rbind(plot1_df, plot3_df)
leg <- ggplot(Qsoilplot_df, aes(x = Months, y = Qsoil,
  col = Ident)) + geom_line(size = 2) + scale_color_manual(values = c("orange",
  "navyblue", rainbow(12), "brown")) + theme_bw(base_size = 65) +
  guides(colour = guide_legend(ncol = 3, override.aes = list(size = 20))) +
  labs(col = "ERA5 Qsoil1 data")
legend <- cowplot::get_legend(leg)
grid.newpage()
grid.draw(legend)

```

A.4.4 Miscellaneous Figures

Chunk 27: Producing miscellaneous figures used for flow charts and scheme overviews.

```

rm(list = ls()) # clearing environment
#####----- PACKAGES -----
source("Y - Codes/S0a_Packages.R") # loading packages
#####----- DIRECTORIES -----
source("Y - Codes/S0b_Directories.R") # setting directories
#####----- FUNCTIONS -----
source("Y - Codes/S0c_Functions.R") # Loading miscellaneous functions

## MEMORY COMPONENT SCHEMATIC -----
cell <- 34881
RegionFile <- "Iberian Region"
From <- 1982
To <- 2015
Lags <- 0:12
Yearvec <- rep(c(1982:2015), each = 12)
Time <- paste("1", (From - 1), "_12", To, sep = "")
DetVars <- c("NDVI", "Qsoil1", "Qsoil2", "Qsoil3", "Qsoil4", "Tair")
Dir.Memory.Reg <- paste(Dir.Memory, "/", RegionFile, "-", From - ceiling(1/12 * max(Lags)),
                         "_", To, sep = "")
# Era5
Environment <- list.files(Dir.ERA.Monthly)
EnvRegion <- Environment[grep(RegionFile, Environment)]
EnvRegion <- EnvRegion[grep(Time, EnvRegion)]
setwd(Dir.ERA.Monthly)
Qsoil1 <- brick(EnvRegion[1])
Qsoil2 <- brick(EnvRegion[2])
Qsoil3 <- brick(EnvRegion[3])
Qsoil4 <- brick(EnvRegion[4])
Tair <- brick(EnvRegion[5])
# NDVI
Vegetation <- list.files(Dir.Gimms.Monthly)
VegRegion <- Vegetation[grep(RegionFile, Vegetation)]
setwd(Dir.Gimms.Monthly)
NDVI <- brick(VegRegion[1])
NDVI <- NDVI[[c(as.numeric(min(which(Yearvec == From)):max(which(Yearvec == To))))]]
setwd(mainDir)
# Memory
Memory <- list.files(Dir.Memory)
Memory <- Memory[grep(RegionFile, Memory)]
Memory <- Memory[grep(paste(From - ceiling(1/12 * max(Lags)), To, sep = "-"), Memory)]
## Plotting Frames
cells <- adjacent(NDVI, cells = cell, directions = 8, include = TRUE)
cells <- cells[10:18]
NDVI_vec <- as.numeric(apply(NDVI[cells], 2, mean))
NDVI_df <- data.frame(NDVI = c(as.numeric(NDVI_vec)))
NDVI_df <- transform(NDVI_df, DeNDVI = c(detrend(NDVI, tt = "linear")))
Data_df <- data.frame(Qsoil1 = as.numeric(apply(Qsoil1[cells], 2, mean)), Qsoil2 = as.numeric(apply(Qsoil2[cells],
                                                2, mean)), Qsoil3 = as.numeric(apply(Qsoil3[cells], 2, mean)), Qsoil4 = as.numeric(apply(Qsoil4[cells],
                                                2, mean)), Tair = as.numeric(apply(Tair[cells], 2, mean)))
# Data Manipulation
Data_df <- transform(Data_df, NDVI = c(NDVI_df$NDVI, rep(NA, 12)), DeNDVI = c(NDVI_df$DeNDVI,
                           rep(NA, 12)), DeQsoil1 = detrend(Qsoil1, tt = "linear"), DeQsoil2 = detrend(Qsoil2,
                           tt = "linear"), DeQsoil3 = detrend(Qsoil3, tt = "linear"), DeQsoil4 = detrend(Qsoil4,
                           tt = "linear"), DeTair = detrend(Tair, tt = "linear")) # linear detrending
Data_df <- as.data.frame(cbind(Data_df[, 6], Data_df[, c(1:5, 7:12)]))

```

```

colnames(Data_df)[1] <- "NDVI"
Data_df$Qsoil1 <- c(Data_df$Qsoil1[c(-1, -2)], NA, NA)
Data_df1 <- Data_df[1:24, ]
## Intrinsic
Intrinsic_df <- data.frame(Data = Data_df1$NDVI, Variable = rep("Response", dim(Data_df1)[1]),
  Time = rep(1:dim(Data_df1)[1], 1))
ggplot(data = Intrinsic_df, aes(x = Time, y = Data, color = Variable)) + geom_line() +
  theme_bw() + geom_point() + scale_colour_manual(values = c("darkgreen")) + ggtitle("Intrinsic Memory") +
  xlab("Time [months]")
## Extrinsic
Extrinsic_df <- data.frame(Data = c(Data_df1$NDVI, Data_df1$Qsoil1), Variable = c(rep("Response",
  dim(Data_df1)[1]), rep("Extrinsic", dim(Data_df1)[1])), Time = rep(1:dim(Data_df1)[1],
  2))
ggplot(data = Extrinsic_df, aes(x = Time, y = Data, color = Variable)) + geom_line() +
  theme_bw() + geom_point() + scale_colour_manual(values = c("blue", "darkgreen")) +
  ggtitle("Extrinsic Memory") + xlab("Time [months]")

## KRIGING SCHEMATIC ---- data
setwd(Dir.ERA.Monthly)
tairf <- brick(list.files()[15])
setwd(Dir.ERA)
tairr <- brick(list.files()[5])
# cropping
Shapes <- readOGR(Dir.Mask, "ne_50m_admin_0_countries", verbose = FALSE)
RegObj <- RegionSelection(Region = c("Portugal", "Spain", "France", "Andorra"), RegionFile = "Iberian Region",
  Extent = extent(-10, 10, 35, 52))
area <- RegObj[[1]]
location <- RegObj[[2]]
RegionFile <- RegObj[[3]]
rasinter <- crop(tairr, area) # cropping to extent
tairr <- mask(rasinter, Shapes[location, ]) # masking
# plotting
col.tair <- colorRampPalette(c("blue", "turquoise", "yellow", "orange", "red"))(100)
plot(tairr[[1]], colNA = "black", main = paste("Tair January 1981"), cex.main = 1.5,
  legend.width = 2, legend.shrink = 1, axis.args = list(cex.axis = 1.5), cex.axis = 1,
  col = col.tair, axes = TRUE, legend = TRUE)
plot(tairf[[1]], colNA = "black", main = paste("Tair January 1981"), cex.main = 1.5,
  legend.width = 2, legend.shrink = 1, axis.args = list(cex.axis = 1.5), cex.axis = 1,
  col = col.tair, axes = TRUE, legend = TRUE)

## MODEL SCHEMATIC ----
load(paste(Dir.Data, "ModData_df.RData", sep = "/")) # model data for pixel 35363 in Iberian rasters
## location plot
IbRas <- mean(brick(paste(Dir.Gimms.Monthly, list.files(Dir.Gimms.Monthly)[10], sep = "/")))
cells <- adjacent(IbRas, cells = 35363, directions = 8, include = TRUE)
cells <- cells[10:18]
Cellras <- IbRas
values(Cellras)[cells] <- 8888
values(Cellras)[which(values(Cellras) != 8888)] <- NA
plot(IbRas, colNA = "black", main = "Exemplatory Raster Cells", cex.main = 1.5, legend = FALSE,
  legend.width = 1, legend.shrink = 1, axis.args = list(cex.axis = 1), cex.axis = 1)
plot(Cellras, col = "red", add = TRUE, legend = FALSE, axis.args = list(cex.axis = 1),
  cex.axis = 1)
## NDVI series
plot_df <- data.frame(Months = rep(1:dim(ModData_df)[1], 2), NDVI = c(ModData_df$`NDVI_anom[1:length(Clim2_anom)]``,
  ModData_df$`NDVI_Lag1[1:length(Clim2_anom)]`), Data = c(rep("Z-Scores", dim(ModData_df)[1]),
  rep("t-1", dim(ModData_df)[1])))
ggplot(plot_df, aes(x = Months, y = NDVI, col = Data)) + geom_line(size = 1.2) +

```

```
theme_bw() + labs(title = "NDVI time series")
## Cummulative soil moisture lags
plot_df <- with(ModData_df, data.frame(Months = rep(1:dim(ModData_df)[1], 1), Qsoil = c(ClimCum_6)))
ggplot(plot_df, aes(x = Months, y = Qsoil)) + geom_line(col = "navyblue", size = 1.2) +
  theme_bw() + labs(title = "Cummulative Soil Moisture (0-7cm) Lag 6")
## Tair
plot_df <- with(ModData_df, data.frame(Months = rep(1:dim(ModData_df)[1], 1), Qsoil = Clim2_anom))
ggplot(plot_df, aes(x = Months, y = Qsoil)) + geom_line(col = "red", size = 1.2) +
  theme_bw() + labs(title = "Air Temperature")
## PCA
biplot(princomp(pca_mat))
```

A.5 Declaration Of Authorship

I, Erik Kusch, hereby declare that this thesis and the work presented in it is entirely my own. Where I have consulted the work of others, this is always clearly stated.

Erik Kusch

Signature

Date