Food and Agriculture Organization of the United Nations

Global Soil Organic Carbon Sequestration Potential Map

GSOCseq V1.1

Technical Manual

Pillar 4 Working Group & **INSII**

GLOBAL SOIL PARTNERSHIP

Global soil organic carbon sequestration potential map (GSOCseq v1.1)

Food and Agriculture Organization of the United Nations

Rome, 2022

Required citation:
Peralta, G., Di Paolo L., Luotto, I., Omuto, C., Mainka M., Viatkin, K., Yigini, Y. 2022. *Global soil organic carbon sequestration potential map (GSOCseq v1.1) – Technical manual.* Rome, FAO. https://doi.org/10.4060/cb2642en

ISBN 978-92-5-133796-7

Cover and back cover photograph:
©FAO/Ronald Vargas

Contents

Figures

Tables

Abbreviations

BAU	Business as usual
BIO	Microbial Biomass
Ceq	Estimated annual C input at equilibrium
Ct	Annual carbon inputs for a specific year
CO2	Carbon dioxide
DM	Dry matter
DPM	Decomposable plant material (C pool)
GHG	Greenhouse gases
GSOCmap	Global Soil Organic Carbon Map
GSOCseq	Global Soil Organic Carbon Sequestration Potential Map
GSP	Global Soil Partnership
HUM	Humified soil organic matter (C pool)
HWSD	Harmonized World Soil Database
INSII	International Network of Soil Information Institutions
IOM	Inert organic matter (C pool)
IPCC	Intergovernmental Panel on Climate Change
ITPS	Intergovernmental Technical Panel on Soils
NDVI	Normalized difference in vegetation index
NPP	Net Primary Production
P4WG	Pillar 4 Working Group
QA/QC	Quality Assurance/Quality Check
RMSE	Root mean square error
RPM	Resistant plant material (C pool)
SOC	Soil organic carbon
SOCmeas	Measured soil organic carbon (as in GSOCmap)
SOCseq	Soil organic carbon sequestration

SOCsim	Simulated soil organic carbon after the first equilibrium run
SOM	Soil organic matter
SSM	Sustainable soil management
SSM1	Low carbon inputs sustainable soil management scenario
SSM2	Medium carbon inputs sustainable soil management scenario
SSM3	High carbon inputs sustainable soil management scenario
VGSSM	Voluntary Guidelines for Sustainable Soil Management

List of contributors

Authors

Guillermo Peralta (GSP Secretariat)
Luciano Di Paolo (GSP Secretariat)
Isabel Luotto (GSP Secretariat)
Christian Omuto (GSP Secretariat)
Moritz Mainka (GSP Secretariat)
Kostiantyn Viatkin (GSP Secretariat)
Yusuf Yigini (GSP Secretariat)

Scientific Advisor

Pete Smith (University of Aberdeen)

Cover design

Matteo Sala

Contributors and Reviewers

P4WG	Pillar 4 Working Group
INSII	International Network of Soil Information Institutions
ITPS	Intergovernmental Technical Panel on Soils
4per1000 SCT	4 per 1000 Scientific and Technical Committee
CIRCASA	Coordination of International Research Cooperation on Soil Carbon Sequestration in Agriculture
UNCCD-SPI	The UNCCD Science-Policy Interface
Thünen Institute	German Federal Research Institute for Rural Areas, Forestry and Fisheries

Chapter 1

Background

Soils have become one of the world's most vulnerable resources in face of climate change, land degradation, biodiversity loss, and increased demand for food production. The role of soils and soil organic carbon (SOC) in the climate system and climate change adaptation and mitigation has been recognized widely and validated in various studies, both experimentally and through modelling. Maintaining and increasing SOC stocks is not only crucial for reducing greenhouse gas (GHG) emissions and removing CO_2 from the atmosphere but also for harnessing the benefits of increased SOC (and soil organic matter – SOM) for soil health and fertility by improving water storage and thereby increasing the access of plants to water, food production potential, and resilience to drought (FAO, 2017). It may be beneficial to human health, decreases the imminent threat of food insecurity to rural communities, and enables us to thrive on our planet. The widespread adoption of site–specific sustainable soil management (SSM) practices in agricultural lands can harness a large C sink capacity at the global scale and its role as an effective strategy to mitigate GHG has been widely documented. The magnitude and rate of carbon sequestration in soils can vary greatly, depending on the different land uses and practices, soil characteristics, vegetation, topography, and climate, among other soil forming factors and processes. It is thus relevant to identify which regions, environments and production systems have a greater potential to increase SOC stocks as well as establishing priorities for research and the implementation of public policies. Responding to a request for support in addressing the Sustainable Development Goal Indicators, especially indicator 15.3 which includes the restoration

of degraded soils, the Global Soil Partnership (GSP) Plenary Assembly in 2020 instructed the Intergovernmental Technical Panel on Soils (ITPS) and the GSP Secretariat to develop the Global Soil Organic Carbon Sequestration Potential map (GSOCseq map), following the same country–driven approach developed for the Global Soil Organic Carbon map (GSOCmap). This 'bottom–up' approach is expected to generate a GSOCseq map from national SOCseq maps, developed and validated by local experts, based on the implementation of SOC models using standardized procedures and by leveraging the best available local data.

To this end, members under the International Network of Soil Information Institutions (INSII) umbrella developed general guidelines and technical specifications for the preparation of the GSOCseq map and countries were invited to prepare their national soil organic carbon sequestration potential maps according to these specifications. Given the scientific advances in the tools for modeling and mapping SOC, many countries requested the GSP Secretariat to support them in the process of preparing national SOCseq maps. Hence, an intensive capacity development program on SOC sequestration potential through the use of modeling and mapping techniques is being implemented. Regional and national training sessions are being organized using an on–the–job–training modality to ensure that national experts are trained on the state of the art modeling and mapping techniques using their own data sets to produce reliable SOCseq maps. This Technical Manual was prepared as a comprehensible reference knowledge base to support the capacity development process. It provides the necessary background knowledge, methodologies and technical steps to produce national SOCseq maps. It includes step–by–step guidance for developing 1x1 km grids of SOC sequestration potential, as well as for the preparation of local data and the compilation and pre–processing of spatial datasets, using open source software environments and tools. This Technical Manual supplements the Technical Specifications and Country Guidelines for Global Soil Organic Carbon Sequestration Potential Map (GSOCseq) (FAO and GSP, 2020), and part of its contents is included in the current Manual. It is our hope that this Technical Manual will fulfill its mandate of easily enabling any user to produce digital SOC sequestration maps using soil legacy data, process–oriented SOC models and modern techniques of digital soil mapping and realize the overall aim of improving decision making on soil management.

1.1 How to use this book

This Manual is organized in 13 Chapters, including this one. Chapter 2 provides scientific background on the importance of SOC sequestration as a GHG mitigation strategy, factors that regulate SOC sequestration and finally how it can be estimated. Chapter 3 provides scientific background on the use of different SOC models to estimate SOC changes and SOC sequestration potential. Chapter 4 describes one of the most used process–oriented SOC models, to be used as a standard to allow comparisons between countries: the RothC model. Chapter 5 describes the general modeling procedures to generate national SOCseq maps using this SOC model. Chapter 6 describes the required datasets, potential data sources and methodologies to estimate modeling inputs for the modeling approach described in Chapter 5. Chapter 7 gives an overview of the steps and software requirements to generate national SOCseq maps using a spatially explicit R–version of the RothC model. Chapters 8 to 11 constitute the core of this document, providing a step by step guide on how to run the R–scripts and generate the final product:

- Chapter 8 summarizes the different steps and scripts to be used during the complete process;
- Chapter 9 summarizes the procedures and scripts harmonization and preparation of input data;
- Chapter 10 summarizes the procedures for running the model from the prepared data;
- Chapter 11 summarizes the procedures for the generation of the final maps from the modeling runs.

Chapter 12 provides an overview for the estimation of uncertainties. Lastly, Chapter 13 summarizes the procedure for the sharing of the national SOCseq maps.

1.2 The sample dataset

The sample dataset provided with this Technical Manual was compiled to model and map the potential soil organic carbon sequestration for the district of Pergamino, Buenos Aires. It covers an area of 2 950 km^2 that is mainly dedicated to agriculture. Pergamino is located in the Corn Belt of Argentina and with its subtropical climate it is favorable for double cropping systems, allowing for two harvests a year. The lowlands of this district are used mainly as pastures for the production of calves.

This Technical Manual used in combination with the provided sample dataset and scripts can be used as a step–by–step guide, which covers data preparation and harmonization, modeling potential soil organic carbon sequestration, rasterizing the results, estimating the uncertainty and finally data sharing.

The sample dataset and scripts can be found in the following repository: https://drive.google.com/drive/folders/1Jonn17LAiCyaUSpUn6W63o16bjTg1P0E?usp=sharing

More information on the necessary data and the provided data set can be found in Chapter 6, while an overview of the different steps and scripts needed is presented in Chapter 8.

Chapter 2

Soil organic carbon (SOC) sequestration

2.1 Soil carbon

Soils constitute the largest terrestrial carbon (C) pool. Total soil carbon (C) stock comprises soil organic C (SOC) and soil inorganic C (SIC) components. SOC is the carbon component of soil organic matter (SOM), a heterogeneous pool of C comprised of diverse materials including fine fragments of litter, roots and soil fauna, microbial biomass C, products of microbial decay and other biotic processes (i.e. such as particulate organic matter), and simple compounds such as sugar and polysaccharides (Jansson *et al.*, 2010). The global SOC stock of ice–free land contains about 1 500–2 400 Pg C (1 Pg = 1 Gt) in the top 1 m, 2 300 Pg C in the top 3 m, and 3 000 Pg C in the soil profiles (Batjes *et al.*, 1996; Scharlemann *et al.*, 2014; Tifafi *et al.*, 2018; Lorenz and Lal, 2018). This represents more than the sum of carbon contained in the atmosphere and vegetation (Smith *et al.*, 2019). Soil inorganic C comprises pedogenic carbonates and bicarbonates, which are particularly abundant in arid regions and in alkaline soils. The SIC stock is estimated at 700–1 700 Pg C in the top 1–m soil layer (Lorenz and Lal, 2018) and is believed to occur predominantly in the deeper layers of temperate soils Although soils contribute to a major share of agricultural greenhouse gas emissions (GHGs), due to the size of the soil carbon

pool, even small increments in the net soil C storage represent a substantial C sink potential (Paustian *et al.*, 2016). Carbon sequestration implies transferring atmospheric CO_2 into long–lived pools and storing it securely so it is not immediately re–emitted (Lal *et al.*, 2018). Thus, soil C sequestration means increasing SOC and SIC stocks through judicious land use and sustainable soil management (SSM) practices. Due to the current knowledge on SOC dynamics, the global distribution, and the current knowledge on size of the SOC pool compared to the SIC pool, this technical manual will focus on SOC sequestration.

2.2 SOC sequestration

The basic process of SOC sequestration in the terrestrial biosphere involves transfer of atmospheric CO_2 into plant biomass and conversion of biomass into stable SOC through formation of organo–mineral complexes (Lal *et al.*, 2018). Thus, soil carbon sequestration relies on plant photosynthesis to carry out the initial step of carbon "removal" from the atmosphere. However, rather than increasing the storage of carbon contained in plant biomass, SOC sequestration relies on management practices that increase the amount of carbon stored as soil organic matter, primarily in cropland and grazing lands. The main advantage of scaling up soil C sequestration as a biological negative emission strategy is that carbon stocks are most depleted on lands currently under agricultural management and thus this approach does not require land use conversions (e.g., to forests) nor it increases the competition for land resources. In addition, increases in SOC stocks are highly beneficial in maintaining and increasing soil health and soil fertility, which provides additional incentives for adopting SOC sequestering practices (Paustian *et al.*, 2019).

2.3 Factors affecting SOC sequestration

SOC sequestration is governed by the balance between the rate of C added to the soil from plant residues (including roots) and organic amendments (e.g., manure, compost), and the rate of C lost from the soils, which is mainly as CO_2 from decomposition processes (i.e., heterotrophic soil respiration). Other forms of organic C can be lost as CH_4 from anaerobic (e.g. flooded) reactions and to a lesser extent through leaching of dissolved organic C. Soil erosion can greatly affect C stocks at a particular location, but at larger scales erosion

may not represent a loss process *per se* but rather a redistribution of soil C (Paustian *et al.*, 2019). Decomposition rates are controlled by a variety of factors including soil temperature and moisture, drainage (impacting soil O_2 status) and pH (Paustian *et al.*, 2019). Soil physical characteristics such as texture and clay mineralogy also impact the longevity and persistence (i.e., mean residence time) of soil C, by affecting organic matter stabilization processes, through mineral–organic matter associations (Schmidt *et al.*, 2011; Paustian *et al.*, 2019). In native ecosystems the rate of C inputs is a function of the type (e.g., annual vs. perennial, woody vs. herbaceous) and productivity of the vegetation, largely governed by climate (mainly temperature and precipitation) but also nutrient availability and other growth determining factors. In managed ecosystems such as cropland and grazing land both the rate of C input as well as the rate of soil C loss via decomposition are impacted by the soil and crop management practices applied. There is no one universal management practice to increase SOC sequestration (Lal *et al.*, 2018), but in general, soil C stocks can be increased by: (a) increasing the rate of C addition to the soil, which removes CO_2 from the atmosphere, and/or (b) reducing the relative rate of loss (as CO_2) via decomposition, which reduces emissions to the atmosphere that would otherwise occur (Paustian *et al.*, 2019). Three key aspects need to be considered regarding the pattern of gains or losses of soil C and hence SOC sequestration (Paustian *et al.*, 2019). The first is that with increased C inputs and/or decreased decomposition rates, soil C stocks tend toward a new equilibrium state and thus after a few decades C gains attenuate, becoming increasingly small over time. Secondly, although sequestered SOC can be highly stable, changes in management that lead to C gains are potentially reversible, i.e., if management reverts back to its previous condition, much or all of the gained C can be lost. Thus, practices that led to increased soil C need to be maintained long term. Third, mineral soils (i.e., non–peat soils) have an upper limit or "saturation level" of soil C (Six *et al.*, 2002). While this maximum soil C concentration is well above the observed C concentration of most managed soils, carbon rich mineral soils that already have very high SOC levels (e.g., >5 percent C by mass) may have a propensity for further C gains.

2.4 Estimating SOC sequestration potential

Taking into account the above mentioned factors, SOC sequestration potential after the adoption of SSM practices under specific conditions can be expressed

in different ways depending on the definition of SOC baseline stocks and time towards a new equilibrium state. This Technical Manual will refer to two types of SOC sequestration: an 'absolute SOC sequestration' (SOCseq abs), expressed as the change in SOC stocks over time relative to a base period (or reference period, t0); and a 'relative SOC sequestration' (SOCseq rel), expressed as the change in SOC stocks over time relative to business as usual practices (Fig. 2.1). Thus, the 'absolute' attainable SOC sequestration can be determined for business as usual (BAU) and SSM practices (See Chapter 5), and can be either positive, neutral or negative:

$$\Delta SOC_{ABS} tCha^{-1} = SOC_{SSM/BAU\ t} - SOC_{t0} \tag{2.1}$$

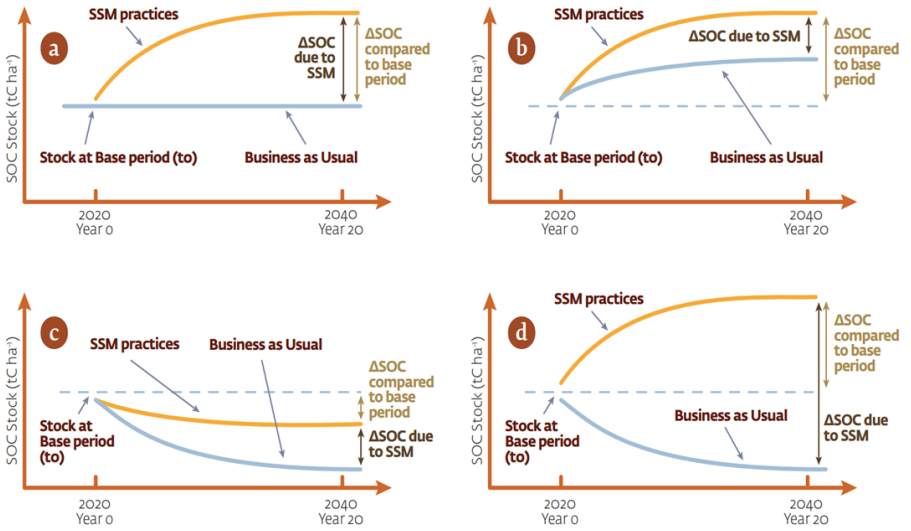
where SOC $SSM/BAU\ t$ refers to the final SOC stocks once a new equilibrium is reached or after a defined period of time (e.g. 20 years), and $SOC\ t0$ refers to the initial or base period SOC stocks (t=0). The 'relative' attainable SOC sequestration is either neutral or positive, can be determined as:

$$\Delta SOC tCha^{-1} = SOC_{SSM\ t} - SOC_{BAU\ t} \tag{2.2}$$

where $SOC_{SSM\ t}$ refers to the final SOC stocks once a new equilibrium is reached or after a defined period of time (e.g. 20 years) after SSM practices are implemented, and $SOC_{BAU\ t}$ refers to the final SOC stocks under business as usual (BAU) practices at the end of the same considered period. Mean annual SOC sequestration rates (t C ha^{-1} yr^{-1}; absolute or relative) can be determined by dividing SOC changes by the duration of the defined period. For more specifications on the approaches proposed in this manual, see Chapter 5.

Thus, agricultural lands may show potential for improvement in their SOC stock after the adoption of SSM practices (compared to business as usual practices), by either gaining or maintaining SOC levels. Four situations are possible: a) lands where SOC levels have reached equilibrium and it is possible to increase levels through SSM; b) lands where the SOC is increasing but can be further increased through SSM; c) lands where SOC is declining and it is possible to stop or mitigate losses in SOC levels through SSM; and d) lands where SOC is declining and it is possible to reverse this fall through SSM. These situations are depicted in Fig. 2.1. It has been estimated that the widespread adoption of site/biome–specific SSM practices can harness a large C sink capacity in agricultural systems at a global scale: 0.4–1.2 PgC/yr (Lal, 2004); 1.0–1.32 PgC/yr (Smith *et al.*, 2008); 0.4–1.1 PgC/yr (De Vries, 2017); 0.32–1.01 PgC/yr (Batjes *et al.*, 2019). However, the extent and rates of SOC sequestration in agricultural

lands may vary greatly depending on the different land uses and practices, soil characteristics, vegetation, topography and climate, among other soil forming factors and processes (Smith *et al.*, 2008; Minasny *et al.*, 2017; Lal *et al.*, 2018; Batjes *et al.*, 2019). Annual sequestration rates due to management practices in croplands and grasslands are usually in the range of 0.2–0.8 tC/ha (Poeplau and Don, 2015; Kampf *et al.*, 2016; Minasny *et al.*, 2017; Conant *et al.*, 2017; Paustian *et al.*, 2016; Paustian *et al.*, 2019). It is therefore relevant to identify which regions, environments and systems have a greater potential to increase SOC stocks and establish priorities for research and implementation of private and public policies. In this sense, coupling SOC models to GIS (Geographic Information Systems) platforms enables the transition t from site–specific SOC stocks estimations to spatial simulations and projections (e.g. Smith *et al.*, 2005; Milne *et al.*, 2007; Kamoni *et al.*, 2007; Falloon *et al.*, 2007; Gottschalk *et al.*, 2012; Lugato *et al.*, 2014), allowing for the identification of conditions that increase the SOC sequestration potential.



SOC theoretical evolutions under business–as–usual (BAU) practices and after the adoption of sustainable soil management (SSM) practices.

Chapter 3

SOC modelling

Modelling is an approach used to infer SOC stocks and distributions in conditions where they have not been measured, such as: under future climatic conditions, at locations or regions where no measurement exists, for management scenarios that have not yet been implemented (FAO, 2019). In the last decades, a number of numerical models have been developed, including mathematical representations that quantitatively describe soil characteristics and processes. The breadth of these approaches can be illustrated by the recent compilation of 90 mathematical models describing SOC changes and biogeochemical related soil processes developed in the last 80 years (Falloon and Smith, 2009; Manzoni and Porporato, 2009; Campbell and Paustian, 2015). However, according to their structure, number of input variables required and temporal and spatial resolution, not all available C models are suitable for all studies (Manzoni and Porporato, 2009).

3.1 Process–oriented models

Among the different types of SOC models, process–oriented multicompartment models have been dominant in efforts to simulate changes in SOC in agricultural lands, grasslands and other production systems (Stockmann *et al.*, 2013). Process–oriented models are built considering the processes involved in the transfer of SOC across the soil profile and its transformations (Smith *et*

al., 1998). They are generally used to predict SOC dynamics based on different conceptual C pools or compartments that alter in size via decomposition rates and stabilization mechanisms (each compartment or pool being a fraction of SOC with similar chemical and physical characteristics; Stockmann *et al.*, 2013). Models belonging to this class can potentially have a variable degree of complexity, from one compartment to multiple compartments (Jenkinson *et al.*, 1990). Early models simulated SOC as one homogeneous compartment (Jenny, 1949). Beek and Frisel (1973) and Jenkinson and Rayner (1977) proposed two–compartment models, and as computational tools became more accessible, multi–compartment models were developed (McGill, 1996).

According to Falloon and Smith (2009), decay rates k are usually expressed in this type of models by first–order kinetics with respect to the concentration C of the pool:

$$\frac{dC}{dt} = -kC \tag{3.1}$$

The flows of carbon within most models represent a sequence of carbon going from plant and animal debris to the microbial biomass and then, to soil organic pools of increasing stability. The output flow from an organic pool is usually split. It is directed to a microbial biomass pool, another organic pool and, under aerobic conditions, to CO_2 . This split simulates the simultaneous anabolic and catabolic activities and growth of a microbial population feeding on one substrate. Two parameters are generally required to quantify the split flow, often defined as a microbial (utilization) efficiency and a stabilization (humification) factor, which control the flow of decayed carbon to the biomass and humus pools, respectively.

3.2 Examples of process–oriented models

CENTURY (Parton, 1996), RothC (Jenkinson *et al.* 1990; Coleman and Jenkinson, 1996), SOCRATES (Grace *et al.*, 2006), DNDC (Li, 1996), CANDY (Franko *et al.*, 1997), DAISY (Hansen *et al.*, 1991), NCSOIL (Hadas *et al.*, 1998) and EPIC (Williams *et al.*, 1983; 1984) are known examples of this kind of process–oriented multicompartment models. They have been developed and tested using long–run data sets, and in general they show a good ability to predict SOC dynamics over decades across a range of land uses, soil types and

climatic regions (Smith *et al.*, 1997). As mentioned before, process–oriented models can be combined with GIS software, giving a modelling platform well suited for global, national, and regional scale studies (e.g. Smith *et al.*, 2005; Milne *et al.*, 2007; Kamoni *et al.*, 2007; Falloon *et al.*, 2007; Gottschalk *et al.*, 2013; Lugato *et al.*, 2014). The review by Campbell and Paustian (2015) emphasizes the fact that among these known process–oriented models, no one clearly outperforms the others. The increase in multi–model comparison publications in the last decades shows the lack of consensus in SOC modelling approaches. It is also noteworthy that among these multi–model comparisons, there was no single model identified with conclusively higher performance capacity. For a detailed comparison between some of the most used SOC models refer to Campbell and Paustian (2015).

However, in order to obtain consistent results in SOC sequestration estimates at a global scale, and to allow comparisons between countries and regions, the use of a standard 'process–oriented' SOC model, following standardized procedures is required in this first step. The Rothamstead soil organic carbon model (RothC; Coleman and Jenkinson, 1996, Chapter 4) is proposed as the standard comparison model in this Technical Manual, for the following reasons:

- It requires less and more easily obtainable data inputs compared to other process–oriented models
- It has been applied using data from long–term experiments across several ecosystems, climate conditions, soils, and land use classes;
- It has been successfully applied at national, regional and global scales; e.g. Smith *et al.* (2005), Smith *et al.* (2007), Gottschalk *et al.* (2012), Wiesmeier *et al.* (2014), Farina *et al.* (2017), Mondini *et al.* (2018), Morais *et al.* (2019);
- It (or its modified/derived version) has been used to estimate carbon dioxide emissions and removals in different national GHG inventories as a Tier 3 approach; according to the latest review by Smith *et al.* (2020): Australia (as part of the FullCam model, Japan (modified RothC), Switzerland, and UK (CARBINE, RothC).

The following Chapter describes the RothC model and its general requirements. Users are nevertheless encouraged to use modified versions of the RothC model (e.g. Farina *et al.*, 2013) if it has been demonstrated that these versions improve estimations under local conditions. Users are also encouraged to provide supplementary SOC sequestration maps developed using alternative preferred SOC

models and procedures. The use of a multi–model ensemble approach (e.g. Riggers *et al.*, 2019; Lehtonen *et al.*, 2020) with selected models is intended for future versions of the GSOCseq map.

Chapter 4

The RothC model

4.1 Model description

RothC is a model for the turnover of organic carbon in non–waterlogged topsoil that allows for the inclusion of the effects of soil type, temperature, moisture content and plant cover on the turnover process, with a monthly time step (Coleman and Jenkinson, 1996). C sequestration in RothC is quantified solely based on soil processes, and as such it is not linked to a plant production model. The user defines carbon inputs to the soil. SOC is split into four active compartments and one inactive compartment which comprises the inert organic matter (IOM). The four active compartments differ in the mean residence time of organic carbon in the soil and are defined as:

- Decomposable Plant Material (DPM);
- Resistant Plant Material (RPM);
- Microbial Biomass (BIO);
- Humified Organic Matter (HUM).

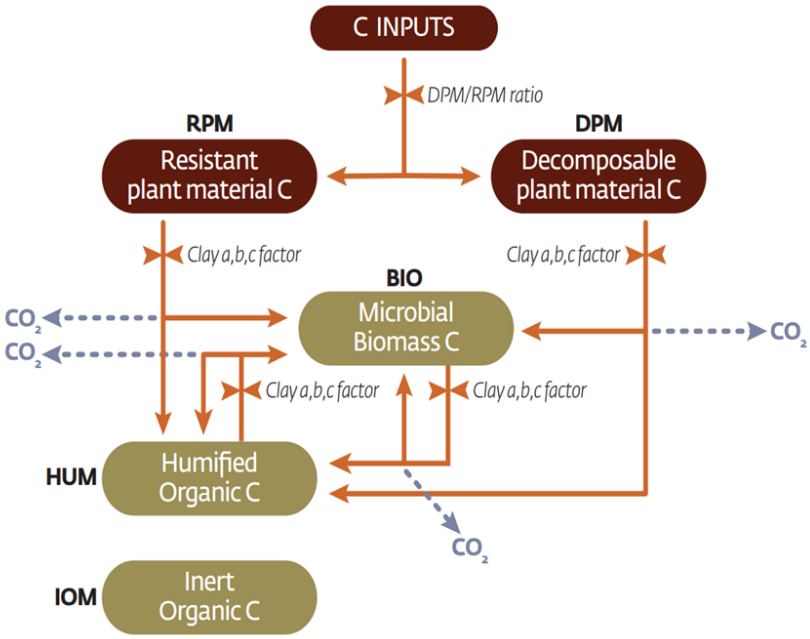
The structure of the model is shown in Figure 4.1. The IOM compartment is resistant to decomposition and is calculated using the following equation (Falloon *et al.*, 1998):

$$IOM = 0.049 \times SOC^{1.139} \tag{4.1}$$

Where SOC is soil organic carbon, tC/ha, and IOM is Inert organic matter, tC/ha. Incoming carbon inputs are split between DPM and RPM, depending on the DPM/RPM ratio of the particular incoming material. For most agricultural crops and improved grassland, the default DPM/RPM ratio is 1.44, i.e. 59 percent of the plant material is DPM and 41 percent is RPM. For unimproved grassland and scrub (including Savanna) a default ratio of 0.67 is used. For a deciduous or tropical woodland a default DPM/RPM ratio of 0.25 is used, i.e. 20 percent of the plant material is DPM and 80 percent is RPM. Both DPM and RPM decompose to form CO_2, BIO and HUM. The proportion that goes to CO_2 and to BIO + HUM is determined by the clay content of the soil. The BIO + HUM is then split into 46 percent BIO and 54 percent HUM. BIO and HUM both decompose to form more CO_2, BIO and HUM. Each compartment decomposes by a first–order process with its own characteristic rate. If an active compartment contains Y tC/ha, this declines at the end of the month to:

$$Ye^{-abckt} tC/ha \tag{4.2}$$

where a is the rate–modifying factor for temperature; b is the rate–modifying factor for moisture; c is the soil cover rate–modifying factor; k is the decomposition rate constant for that compartment; and t is 1/12, since k is based on an annual decomposition rate. $Y(1 - e^{-abckt})$ is the amount of the material in a compartment that decomposes in a particular month. RothC has also been adapted to simulate N and S dynamics (Falloon and Smith, 2009), but nutrient and C dynamics are not interconnected in RothC. It was originally developed and parameterized to model the turnover of organic C in arable topsoil, and it was later extended to model turnover in grasslands, savannas and woodlands, and to operate in different soils and under different climates (Coleman and Jenkinson, 1996).



RothC model structure. Pools and flows of Carbon including major factors controlling the fluxes. Redrawn from Coleman and Jenkinson (1996) and Falloon and Smith (2009).

4.2 RothC general data requirements

The model requires climatic, soil and management data that are relatively easy to obtain or estimate. Each modeling unit (e.g. cell of a grid) requires the following minimum data:

- **Climate data**
  1. Monthly rainfall (mm)
  2. Average monthly mean air temperature (ºC)
  3. Monthly open pan evaporation (mm)/evapotranspiration (mm)
- **Soil data**

1. Total initial 0–30 cm SOC stocks (tC/ha)
2. Initial C stocks of the different pools (tC/ha): DPM, RPM, BIO, HUM, IOM
3. Clay content (percent) at simulation depth
4. DPM/RPM ratio

- **Land–use/Management data**

1. Monthly Soil cover (binary: bare vs. vegetated)
2. Irrigation (to be added to rainfall amounts)
3. Monthly Carbon inputs from plant residue (aboveground + roots + rhizodeposition)
4. Monthly Carbon inputs from organic fertilizers and grazing animals' excretion (tC/ha)
5. Estimate of the decomposability of the incoming plant material

Careful harmonization of modeling procedures, datasets and input estimation methodologies is essential to obtain consistent SOC sequestration results across regions and countries. The general approach and modeling procedures to generate national SOCseq maps using the RothC model are described in Chapter 5. The land use datasets required for the proposed procedures are described in Chapter 6.

Chapter 5

Modeling approach for the GSOCseq

5.1 General framework

SOC sequestration estimates will focus on croplands and grazing lands for the current GSOCseq map version. As defined by IPCC (2006, 2019), croplands include: all annual and perennial crops (cereals, oils seeds, vegetables, root crops and forages); perennial crops (including trees and shrubs, orchards, vineyards and plantations such as cocoa, coffee, tea, oil palm, coconut, rubber trees, bananas, and others), and their combination with herbaceous crops (e.g., agroforestry); arable land which is normally used for cultivation of annual crops, but which is temporarily used for forage crops or grazing as part of an annual crop–pasture rotation (mixed system), is to be included under croplands. Grazing lands include different land uses permanently dedicated to livestock production with a predominant herbaceous cover, including intensively managed permanent pastures and hay land, extensively managed grasslands and rangelands, savannas, and shrublands. Since the proposed standardized methodology and the defined model are neither parameterized nor recommended for use on organic, sandy, saline, and waterlogged soils, soils with SOC stocks higher than 200 tC/ha, sand contents higher than 90 percent and/or electrical conductivity higher than 4 dS/m at 0–30 cm depth, paddy rice lands, peatlands and wetlands

will be masked out from the global results in this first version. Excluded conditions and land uses can be included in future versions of the GSOCseq map, as harmonized procedures for specific conditions are developed. Countries are nevertheless encouraged to provide supplementary maps developed using preferred alternative SOC models and methodologies, especially for excluded conditions.

5.2 Potential SOC sequestration after implementing SSM practices

In order to assess the SOC sequestration potential, SOC stocks in 0–30 cm of mineral soils shall be projected using the RothC model over a 20–year period, under business as usual (BAU) land use and management, and after adoption of SSM practices in croplands and grazing lands (See Chapter 2). A 20–year period is assumed to be the default period during which SOC stocks are approaching a new steady state, to be able to compare results among regions and countries, and with other estimation methods (e.g. IPCC, 2006 Tier 1–2; IPCC, 2019). For some systems, it is acknowledged that the new steady state may take much longer, even more than 100 years, depending on soil and climate characteristics (e.g. Poulton et al, 2018). Together with the 20–years projection, countries can project SOC stocks over 50 or 100 years or more, and determine the stocks and the period at which a new steady state is attained according to local conditions, and produce additional sequestration maps (See mandatory and optional products, Technical Specifications, sections 4.1 and 4.2). As stated in Chapter 2, SOC sequestration potential after the adoption of SSM practices in current agricultural lands shall be estimated by: an 'absolute SOC sequestration' (SOCseq abs), expressed as the change in SOC stocks over time relative to a base period (or reference period, t0); and a 'relative SOC sequestration' (SOCseq rel), expressed as the change in SOC stocks over time relative to the business as usual scenario. Absolute and relative sequestration and sequestration rates for a 20–year period shall be estimated following the equations described in Chapter 2 (eq. 2.1 and 2.2)

5.3 Business as usual and SSM scenarios

SOC stocks in 0–30 cm of mineral soils in current agricultural lands shall be projected over a 20–year period, under a business as usual scenario (BAU) and under sustainable soil management (SSM) scenarios. The BAU scenario refers to the land use, land management, production practices or technologies that are currently being implemented (as in time = 0, or 2020) in croplands and grazing lands. BAU practices represent typical, prevailing practices in a specific agro–ecological zone and productive system. SSM practices refer to management practices that are expected to remove CO_2 from the atmosphere and retain it as SOC, to enhance SOC accumulation, or to mitigate or reverse SOC losses compared to the BAU (See Fig. 2.1). Although there is no universal soil management practice, basic principles are widely applicable, such as those identified in the Voluntary Guidelines for Sustainable Soil Management (VGSSM; FAO, 2017) for increasing soil carbon inputs to soil and enhancing soil organic matter content:

- increasing biomass production and residue returns to the soil;
- using cover crops and/or vegetated fallows;
- implementing a balanced and integrated soil fertility management scheme;
- implementing crop rotations, combining legumes and pulses with high residue crops, or improving the crop–mix;
- effectively using organic amendments, manure, or other carbon–rich wastes (which are not currently applied to soils);
- promoting agro–forestry and alley cropping;
- managing crop residues and grazing to ensure optimum soil cover; among others.

A very wide range of management practices are currently being implemented and can potentially be introduced into the world's agricultural systems, depending on climatic, soil, socio–cultural and economic conditions. In turn, different SSM C–oriented practices are often combined, making it difficult to dissociate their effects on SOC dynamics. Thus, as a first step, and to harmonize the results on a global map, and because soil carbon turnover models are the most sensitive to carbon inputs (FAO, 2019), these guidelines propose to group SSM practices into three scenarios as a standard method, based on their expected relative effects on C inputs compared to BAU: Low, Medium and High increase in C inputs (referred as SSM1, SSM2, and SSM3 scenarios; for technical procedures,

refer to section 5.4). National experts' opinion and local data are essential to accurately estimate or validate the target areas and carbon input levels for the different SSM scenarios in forthcoming versions.

5.4 General modeling procedures

The modeling approach proposed in the current Technical Manual is based on the studies by Smith *et al.* (2005; 2006; 2007), Gottschalk *et al.* (2012) and Dechow *et al.* (2019). The following sections describe the different modeling phases of the approach.

5.4.1 Initialization: spin up

Prior to the simulation of SOC stocks and sequestration under the different scenarios, model initialization is required to set the initial SOC condition (total SOC and partition of the different pools) at the start of the simulation period, and to adjust the C inputs estimates. This modeling phase is referred to as initialization or ´spin up´ through this document. This is a key step, as the outputs of this phase will be used as inputs for the next modeling phase (see sections 5.4.2 and 5.4.3).
Two methodologies to estimate initial carbon pools and initial C inputs are provided:
1. Initialization based on equilibrium runs
2. Initialization based on an analytical approach

5.4.1.1 Spin up phase: initialization with equilibrium runs

This approach is based on studies by Smith *et al.* (2005; 2006; 2007), Gottschalk *et al.* (2012) in large scale studies. The standard procedure is to have a spin up period to initialize the model, so the soil carbon pools are in approximate equilibrium with the initial conditions regarding soil and climate variables, vegetation and land management. The length of the spin up simulation period needed to approach a steady state pool distribution can usually vary between 100s to 1 000s years (FAO, 2019). The C input is adjusted so that the modelled final SOC of this period, and hence initial SOC of the following phases, matches a known SOC stock. In a first initialization step, RothC shall be run iteratively

to equilibrium to calculate the size of the SOC pools and the annual plant carbon inputs using constant environmental conditions (Phase 1, Figure 5.1), for each grid cell on the map. Ideally, a first equilibrium run for a standard 10 000–year period should be performed, considering constant climatic conditions as the average of historic climate data from 1980 to 2000 (see Chapter 6, Climate data sets), clay contents (see Chapter 6, soil data sets), and land use as in year 2000 (see Chapter 6, land use data sets). Due to the simulated time span and depending on the size of the target area, this modelling phase is the most time consuming and computationally demanding . The duration of the equilibrium run can be reduced if the data suggests that the equilibrium is reached with fewer iterations. A minimum of 500 years is suggested to approach equilibrium with reduced computational time to generate national maps. However, it must be noted that spin up runs for 500 years may not necessarily end up in equilibrium SOC stocks, depending on soil, climate and land use conditions. Increasing the duration (1 000–2 000 years) will reduce deviations with the cost of additional computation time. The total annual plant C input can be initially assumed to be 1 t C ha^{-1} yr^{-1} and the proportions of plant material added to the soil for each month are set to describe the typical input pattern for each land use class (Smith *et al.*, 2007; Mondini *et al.*, 2017). After the first equilibrium run, the annual C inputs from plant residues need to be optimized so that the results of the spin up phase fit with the estimates of total SOC stocks of 0–30 cm provided in the FAO–ITPS GSOCmap. C equilibrium inputs can be adjusted using the following equation (Smith *et al.*, 2005):

$$C_{eq} = C_i \times [\frac{SOC_{GSOCmap} - IOM}{(SOC_{sim} - IOM}] \tag{5.1}$$

where C_{eq} is the estimated annual C input at equilibrium, C_i is the initial annual C addition (the sum of the proportions of the C input in the first equilibrium is 1), $SOC_{GSOCmap}$ is the estimated soil C given in FAO–ITPS GSOCmap, SOC_{sim} is the simulated soil C after the first equilibrium run, and IOM is the C content of the inert organic matter fraction in the soil (all in tC/ha). The size of the IOM fraction (tC/ha) can be set according to the equation given by Falloon *et al.* (1998):

$$IOM = 0.049 \times SOC_{GSOCmap}^{1.139} \tag{5.2}$$

A second long term (minimum 1 000 years) equilibrium run shall be performed

using the estimated C_{eq}, (under the same conditions as the first run), in order to obtain the size of the different SOC pools (tC/ha) at year 2000.

An alternative to further reduce computational time is to avoid this second run by estimating the size of the different SOC pools using pedotransfer functions (Weihermüller *et al.*, 2013). The R implementation the spin up phase can be found in Chapter 10 (section 10.1). The equilibrium run is a widely known approach to initialize RothC and other SOC models (FAO, 2019), and it has been implemented in other global and regional modeling–mapping studies to analyze SOC dynamics (e.g. Smith *et al.*, 2005; 2006; 2007; Gottschalk *et al.*, 2012).
Compared to the analytical approach, which is presented in this Technical Manual as an alternative spin up approach, the equilibrium run allows for further user–defined modifications such as running the model under non–homogeneous conditions (e.g. not constant climatic conditions, land use and management for a specified time period). The approach can be also used to estimate the required period to attain equilibrium SOC stocks under certain environmental conditions, among other relevant research questions. However, although users may be in general more familiar with this initialization approach, it can be considerably time consuming as well as computationally demanding, depending on the simulation area. If homogeneous environmental conditions are assumed during the spin up phase, other approaches (see following section) may be the preferred option.

5.4.1.2 Spin up phase: initialization by analytical solution

Based on the Introductory Carbon Balance Model (ICBM B2) in Kätterer and Andren (2001) and pool–specific differential equations for the RothC model in Sierra and Müller (2015), Dechow *et al.* (2019) developed an analytical solution of RothC which describes the topsoil SOC development assuming temporal homogeneous climatic and management conditions. This novel approach allows quantification of pool distribution and C input for RothC at equilibrium. The structure of the approach is based on the linear relationship between C input amounts and initial SOC that follow from the analytical solution of the RothC model. Under homogeneous conditions, the SOC at time t is linearly correlated to the initial SOC (C_0) and the carbon input rate I (equation 5.3):

$$C(t) = S_{impl}C_0 + \sum_{i=1} NI_i(u_{DPM}\gamma_{DPM_i} + u_{RPM}Y_{RPM_i} + u_{hum}\gamma_{hum_i})C_0 f_{IOM}[Mg\ C\ ha^{-1}] \quad (5.1)$$

Where C_0 is the initial SOC stock (which corresponds to $SOC_{GSOCmap}$, the estimated soil C given in FAO GSOCmap). S_{impl},u_{DPM}, u_{RPM} and u_{hum} are functions integrating model structure and parametrization of RothC. Parameters γ_{DPM}, γ_{RPM}, γ_{HUM} are the partition coefficients. These γ coefficients will depend on the decomposability of the incoming residues. For example, in conditions with carbon inputs with a DPM/RPM of 1.44, the γ DPM equals 0.59 and γ DPM equals 0.41 . N is the number of input substrates characterized by a specific set of partition coefficients and f_{IOM} is the fraction of inert SOC (IOM, equation 5.2). For stationary conditions time is assumed to be infinite and therefore the effect of initial active SOC (C_0 - IOM) negligible (equarion 5.4):

$$C_0 = \sum_{i=1}^{N} I_i(\gamma_{DPM_i}u_{DPM} + \gamma_{HUM_i}u_{HUM}) + f_{IOM}C_0 \quad (5.4)$$

First, the fractions f_i of the DPM, RPM, BIO and HUM pools at equilibrium are estimated following the set of equations described in in the supplementary material of Dechow *et al.* (2019) (https://www.fao.org/fileadmin/user_upload/GSP/GSOCseq/supplementary_material_analytical_spinup.pdf). The estimated fractions of each SOC pool at equilibrium will depend on:

- the decomposition rates constants (k) of the different carbon pools
- an average of the different modifying factors (temperature, soil moisture, vegetation factors)
- initial SOC stock (total SOC stock at equilibrium) clay content
- the product ratio CO_2/decomposed C remaining (depending on clay content)
- the ratio of C fluxes to BIO and HUM partition coefficients of the C input (DPM/RPM ratio)
- IOM fraction compared to total C

These equations simplify when assuming an infinite time t (equilibrium). Equations 5.5–5.8 quantify the fraction of each C pool related to the active C (C_0 - IOM):

$$f_{DPM} = \frac{u_{DPM}\gamma_{DPM}}{u_{DPM}\gamma_{DPM} + u_{RPM}\gamma_{RPM} + u_{HUM}\gamma_{HUM}} \tag{5.5}$$

$$f_{RPM} = \frac{u_{RPM}\gamma_{RPM}}{u_{DPM}\gamma_{DPM} + u_{RPM}\gamma_{RPM} + u_{HUM}\gamma_{HUM}} \tag{5.6}$$

$$f_{BIO} = \frac{u_{BIO\ DPM}\ \gamma_{DPM} + u_{BIO\ RPM}\gamma_{RPM} + u_{BIO\ HUM}\ \gamma_{HUM}}{u_{DPM}\ \gamma_{DPM} + u_{RPM}\ \gamma_{RPM} + u_{HUM}\ \gamma_{HUM}} \tag{5.7}$$

$$f_{HUM} = \frac{u_{HUM\ DPM}\gamma_{DPM} + u_{HUM\ RPM}\gamma_{RPM} + u_{HUM}\gamma_{HUM}}{u_{DPM}\gamma_{DPM} + u_{RPM}\gamma_{RPM} + u_{HUM}\gamma_{HUM}} \tag{5.8}$$

$$f_{IOM} = \frac{IOM}{C_0} \tag{5.9}$$

Where γ are the partition coefficients of the incoming carbon inputs as explained in equation 5.3, and u coefficients are the result of functions integrating model structure and parametrization of RothC, following the equations in the supplementary material of Dechow et el. (2019) (https://www.fao.org/fileadmin/user_upload/GSP/GSOCseq/supplementary_material_analytical_spinup.pdf). Once the fractions of the different pools are estimated, the amount of Carbon (tC/ha) in each pool is estimated from the total and active (other than IOM) SOC stocks:

$$SOC_{active} = SOC_{GSOCmap} - IOM \tag{5.10}$$

$$SOC_{pool_i} = f_{pool_i} \times SOC_{active} \tag{5.11}$$

where SOC_{active} represents the SOC stocks of all the active pools of RothC model in tC/ha (DPM, RPM, BIO and HUM), IOM represents the Inert Organic Carbon estimated from equation 5.2, $SOC_{GSOCmap}$ is the estimated soil C given in FAO–ITPS GSOCmap in tC/ha (representing the total SOC stocks at equilibrium), SOC_{pool_i} represents the SOC stock of each of the active pools in tC/ha, and f_{pool_i} represents the fraction of each active pool estimated by the analytical procedure and equations 5.5 to 5.8. Finally, Carbon inputs (C_i) at equilibrium can be estimated as:

$$C_i = \frac{SOC_{GSOCmap} - IOM}{\gamma_{DPM} u_{DPM} + \gamma_{RPM} u_{RPM} + \gamma_{HUM} u_{HUM}} \tag{5.12}$$

Pool distributions and equilibrium C input quantification can be more accurate (closer to equilibrium) and is computationally faster with the analytical solution. The complete R implementation of this procedure can be found in Chapter 10 (section 10.1).

5.4.2 Warm up

Since FAO GSOCmap SOC was generated from individual SOC measurements taken over different decades (i.e. 1960s to 2000s), a temporal harmonization of SOC stocks can be performed as a second initialization step to minimize differences in current SOC stocks at year 0 (i.e. initial SOC stocks at year 2020), and account for climatic variations in the 2000–2020 period:

- SOC stocks from the GSOCmap shall be considered to be the stocks twenty years prior to the simulation (t = -20 yr; i.e. year 2000).
- A 20–year 'short spin up' run can be performed to adjust for major deviations among different measurement periods on the GSOCmap (figure 5, Phase 2), using year–to–year climatic conditions for the period 2001–2020 (See Chapter 6, Climate data sets), clay contents (See Chapter 6, soil data sets), the stocks in the different SOC pools from the results of the 'long spin up' run, and land use as in year 2020 (land use representative of the period 2001–2020; or yearly land use data shall be used when available).
- Year–to–year C inputs over the period 2001–2020 should be adjusted considering year–to–year changes in estimated Net Primary Production (NPP), (details in Chapter 6, monthly carbon inputs). SOC stocks can either increase or decrease during this 'short spin up' stage.

This 'short spin up' period is intended to: reduce the effects of different time measurements in the GSOCmap (over– or underestimation of current initial SOC stocks); minimize initialization effects (e.g. deviations in the estimation of initial pool sizes); and account for the effects of sub–regional, regional and global climatic and land use changes over the period 2001–2020 and their effects on NPP. If recent (2015–2020) national SOC monitoring campaigns have been undertaken to generate the latest version of the FAO–ITPS GSOCmap, the SOC

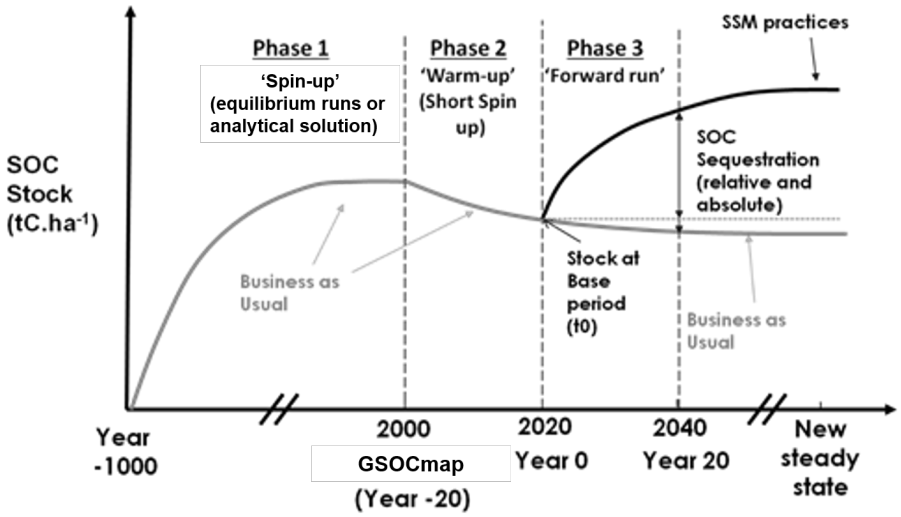
stocks from the GSOCmap can be considered as the current stocks (t = 0 y; i.e. year 2020), and the 'short spin up' phase is not required.

5.4.3 Forward runs

After the equilibrium and 'short spin up' runs, SOC sequestration due to SSM practices can be estimated in a forward run (Figure 5.1, phase 3). SOC stocks can be simulated from 2020 (t=0) to 2040 (t = +20) for the BAU and the three SSM scenarios, using average mean monthly climate variables (2001–2020), C inputs adjusted as described in Chapter 6 and land use maps from 2020. It should be noted that global climatic changes are to be expected over the next 20 years. However, climate change projections diverge significantly in the second half of the century, after the year 2050 (IPCC, 2014; 2018). As there is a lack of consensus over which climate projections to use for future scenarios as well as a significant divergence in terms of climatic trends after 2050, the use of monthly average climatic variables from 2001–2020 for the period 2020–2040 is set as the standard for the forward run. However, the proposed methodology allows for the integration of climate change scenarios, especially for longer–term projections (i.e. + 2050) in future versions.
The absolute SOC sequestration is estimated as the difference between the corresponding SOC stocks from the forward modeling at year +20 (2040) for the different scenarios and the estimated baseline SOC stocks for year 0 (year 2020; refer to equation 2.1). The relative SOC sequestration is to be determined as the difference between the corresponding SOC stocks from the forward at year +20 (2040) for the SSM scenarios and the simulated SOC stocks at year +20 (2020) for the BAU scenario (refer to equation 2.2).

5.5 Summary

The different modeling phases and their data requirements are summarized in in Figure 5.1 and Table 5.1.



The different data sets required to run the RothC model for the different modeling phases are described in Chapter 6.

Summary of the modeling phase 1 and data requirements

Variables	Phase 1 Long spin up Equilibrium
Time span	Minimum 500 years (using equilibrium runs procedure) Infinite (Analytical solution procedure)
Climatic inputs	1980-2000 series monthly average: Rain, Temperature, Evaporation/ Evapotranspiration
Soil inputs	Topsoil clay content
Initial SOC stocks and pools	Inert organic matter (IOM) as determined by equation 5.2 "= 0" for all other fractions (when using equilibrium runs)
Carbon inputs	First run : 1tC.ha-1 Adjusted C inputs from equation 5.1 (using equilibrium runs) From equation 5.12 (using analytical solution)
Vegetation cover	Monthly cover determined: by expert opinion, NDVI 2000-2020 or preferred spectral index (see section 3.3.4)
Land Use	Representative land use of the 1980-2000 period (or layer for year 2000; or best available layer)
Modeled Scenarios	BAU
Expected Results	C inputs at equilibrium Total SOC and SOC pools at year t= -20 (2000)

Summary of the modeling phase 2 and data requirements

Variables	Phase 2 Short spin up
Time span	20 years
Climatic inputs	2001-2020 year to year monthly data: Rain, Temperature, Evaporation/ Evapotranspiration
Soil inputs	Topsoil clay content
Initial SOC stocks and pools	Inert organic matter (IOM, determined by eq. 5.2) Other fractions equal to final SOC pools modeled in Phase 1
Carbon inputs	NPP year-to year adjusted C inputs, from equation 7
Vegetation cover	Monthly cover determined: by expert opinion, NDVI 2000-2020 or preferred spectral index (see Ch. 3.3.4)
Land Use	Year to year Land use 2000-2020 (or representative land use of the period; or best available layer)
Modeled Scenarios	BAU
Expected Results	Total SOC and SOC pools at year t=0 (2020)

Summary of the modeling phase 3 and data requirements

Variables	Phase 3 Forward modeling
Time span	20 years
Climatic inputs	2001-2020 series monthly average: Rain, Temperature ,Evaporation/ Evapotranspiration
Soil inputs	Topsoil clay content
Initial SOC stocks and pools	Inert organic matter (IOM) as determined by equation 5.2 Other fractions equal to the final SOC pools modeled in phase 2
Carbon inputs	NPP year-to year
	adjusted C inputs for the BAU, from equation 7 Estimated from % increase vs. BAU for SSM scenarios
Vegetation cover	Monthly cover determined: by expert opinion, NDVI 2000-2020 or preferred spectral index
	(see Ch. 3.3.4)
Land Use	Last available land use layer (e.g. 2015, 2018; 2020) (or best available layer)
Modeled Scenarios	BAU
	SSM Low SSM Medium SSM High
Expected Results	Total SOC and SOC pools at year t=+20 (2040) for the BAU, and SSMs scenarios
	Absolute and relative total sequestration (3 SSMs) Absolute and relativesSequestration rates (3 SSMs)

Chapter 6

Data sets and inputs

A careful harmonization of data sets and input estimation procedures is essential to obtain consistent results across regions and countries. This Chapter describes the required data sets, the potential data sources and the methodologies to estimate required inputs for the modeling approach described in Chapter 5. Procedures for the preparation and harmonization of input data are explained in Chapter 9.

6.1 Climate data sets

Gridded climate data shall be obtained from:

a.) National Sources or a preferred regional data source;
b.) Global data sets, when national or regional gridded historical climate data sets are not available.

The latest version of this Technical Manual has identified and recommends the TerraClimate dataset as an improved global alternative. TerraClimate is a dataset of monthly climate and climatic water balance for global terrestrial surfaces from 1958–2019. It has a monthly temporal resolution, a ~4x4 km spatial resolution and was created by combining high–spatial resolution climatological normals from the WorldClim data set, with coarser spatial resolution, but

time–varying data from Climate Research Unit (CRU) Ts4.0 and the Japanese 55–year Reanalysis (JRA55) (Abatzoglou *et al.*, 2018).

The following variables and data sets are required to run the model (See Chapter 5, General modeling procedures):

- Monthly average air temperature (°C),
- Monthly precipitation (mm),
- Monthly potential evapotranspiration (Penman–Monteith; mm)
- data sets: 1981–1990 (series average); 1991–2000 (series average); 2011–2010 (year to year); 2011–2018 (year to year).

The same data sources must be used in all modeling phases.

6.2 Soil data sets

6.2.1 Initial total SOC stocks

Initial total SOC stocks to 30cm depth (in tC/ha) are to be derived from the GSOCmap (30 arc seconds; ~ 1 km x 1 km resolution grid), latest revised version (FAO–ITPS, 2019). Countries wishing to include an updated or improved estimate of current SOC stocks, compared to the latest version of the GSOCmap, are encouraged to submit their updated national SOCmap to the GSP Secretariat and use it for modeling. Since the GSOCmap was generated from national measurements taken between the 1960s and the 2000s, and no temporal corrections have been developed in many countries, GSOCmap values will represent SOC stocks for the year 2000. A 'short spin–up' model run (20 years) with climate variables and management forcing for the period 2000–2020 shall be performed to reduce the effect of temporal deviations. Thus, the simulated SOC content at 2020 after the 'short spin–up' run will represent the initial SOC stocks prior to implementation of SSM practices (See Chapter 6, General modeling procedures). If recent national SOC monitoring campaigns (2015–2020) have been undertaken to generate the latest version of the FAO–ITPS GSOC map, the SOC stocks from the GSOCmap can be considered as the current stocks (t = 0 y; i.e. year 2020), and the 'short spin–up' phase is not required.

6.2.2 Initial C pools

The initial C stocks in the different pools (in tC/ha) considered in the RothC model (DPM, RPM, BIO, HUM and IOM, Fig. 4.1) shall be estimated following the 'long spin–up' and 'short spin–up' procedure described in Chapter 6.

6.2.3 Soil texture: clay content

The average clay contents over 0–30 cm depth is to be obtained from gridded data (raster format) from:

- national Sources (1 km x 1 km resolution);
- global data sets, where national or regional data sets are not available.

The topsoil clay content (0–30 cm, percent mass fraction; 1 km x 1 km resolution) from the Harmonized World Soil Database (HWSD) or SoilGrids developed by the– International Soil Reference and Information Centre (ISRIC) (see Table 6.2) shall be used as the standard global database if national or regional data is not available in the required format or resolution. Clay content can be averaged at finer resolutions to obtain 1 km x 1 km grids. However, countries are encouraged to produce their own texture and clay content maps to be used as inputs for the SOCseq map, following the digital soil mapping approaches described in the GSOCmap Cookbook (FAO, 2018). Average clay contents over a 0–30 cm depth interval can be derived by taking a weighted average of the predictions over the depth interval using numerical integration (Hengl *et al.*, 2017).

$$\frac{1}{b-a}\int_a^b f(x)dx \approx \frac{1}{(b-a)}\frac{1}{2}\sum_{k=1}^{N-1}(x_{k+1}-x_k)(f(x_k)+f(x_{k+1})) \tag{6.1}$$

where N is the number of depths; b is 30 cm, a is 0 cm, xk is the k–th depth and f(xk) is the value of the target variable (i.e., clay content) at depth xk. For example, for the 0–30 cm depth interval, with soil clay values at the first four standard depths (0, 5, 15 and 30 cm) equal to 14.5, 25.0, 25.3 and 25.0, clay content 0–30 cm equals:

$$\frac{(5-0)\times(14.5+25.0)+(15-5)\times(25.0+25.3)+(30-15)\times(25.3+25.0)}{30\times 0.5} = 24.25 \quad (6.1)$$

6.3 Land cover data sets

The gridded land cover data layers shall be obtained from:

- national or regional sources;
- global data sets, where national or regional land use or land cover data sets are not available.

Since land cover may vary substantially between data sources and estimates of past and current land cover may have important deviations from real land cover and land use, users should estimate land use from the source that best reflects national and subnational conditions. Land cover data sets should cover the 2000–2020 (or approximate) period. The ESA (European Space Agency) land cover Global dataset (See Table 6.2), and its reclassification into FAO Global Land Cover – SHARE (GLC–SHARE; See Table 6.2) classes will be provided by the GSP Secretariat, if no national land use dataset is available. However, users should estimate land use from the source that best reflects national and subnational conditions. Other global and regional data sets are provided in Table 6.2 at the end of this Chapter. The land cover classes will affect the decomposability of the incoming plant material (DPM/RPM ratio; See Section 6.6). A spatialized R–version of RothC is provided by the GSP Secretariat (See Chapter 7, software environment) and runs considering the 13 classes defined in the FAO Global Land Cover – SHARE (GLC–SHARE). A default DPM/RPM value is assigned to each class (Table 6.1). Thus, when using this spatialized R–version of RothC without modifying its scripts, the land use classes from the possible different data sets need to be re–classified into FAO Global Land Cover – SHARE (GLC–SHARE) land use classes. However, users can model alternative land use classes, and modify these default DPM/RPM values. If so, modifications in the R–version must be then introduced (See Chapter 7). Examples of land cover reclassification from the ESA land cover database into

the RothC land use categories are presented (Table 6.4) at the end of this Chapter.

FAO Land cover classes, land cover number and default DPM/RPM ratios. "Tree-crops" is shown as an example of the disaggregation of a land use class.

FAO Land aggregated cover class	Land cover code	Default DPM RPM ratio
Artificial surfaces	1	/
Cropland	2	1.44
Grassland	3	0.67
Tree covered	4	0.25
Shrub covered	5	0.67
Herbaceous vegetation	6	0.67
Mangroves	7	/
Spare vegetation	8	0.67
Bare soil	9	/
Snow and Glaciers	10	/
Water bodies	11	/
Cropland-Tree crops*	12	1.44
Paddy fields	13	1.44

As a minimum, land use for the year 2000 and land use for the year 2020 (or last available year) at 1x1 km resolution shall be defined. The predominant land use category in each cell of the 1x1 km grid shall be selected if finer resolutions are available.

6.4 Monthly vegetation cover

It is required to indicate the approximate annual distribution of monthly vegetation cover for the simulations in order to:

- adjust the topsoil moisture deficit estimations (See Chapter 4, Fig. 4.1);
- consider the effects of soil cover on SOC decomposition rates (See Chapter 4, Fig.4.1).

The annual distribution of vegetation cover can be derived from:

- public statistics of national and/or administrative units considering the predominant agricultural systems in a temporal series (2000–2020);
- derived from NDVI (normalized difference in vegetation index) values from historic satellite images (See data sets, Table 6.2).

The occurrence of plant cover can be assumed to be constant in grasslands, shrublands and savannas and during specific months (e.g. 1–6 months for croplands) (e.g. Smith *et al.*, 2005; Smith *et al.*, 2007). The following coefficients can be set for based on the specific land cover and/or land use:

- Perennial tree–crops, forests and grasslands (c=0.6);
- Months with predominantly bare soil and unvegetated fallows (c=1);
- Annual crops (c=0.6).

Considering a temporal series (2000–2020), the proportion of images with NDVI values greater than a specified threshold, indicating active vegetation growth, can be estimated (e.g. NDVI > 0.6). The monthly probability of being vegetated (P veg) can be estimated for each cell grid and each month of the year (1–12), as:

$$P_{veg} = \frac{Number\ of\ images\ NDVI > 0.6}{Total\ images} \tag{6.3}$$

NDVI is proposed as an alternative for estimating vegetation cover when no vegetation cover data or local knowledge is available. The proposed threshold may vary according to local conditions. Global monthly vegetation cover data sets estimated by NDVI (2000–2020) will be provided by the GSP Secretariat. However, NDVI may be a biased indicator in areas with low vegetation cover (e.g. drylands, shrublands), or high nubosity. In these cases, countries are encouraged to use other locally validated spectral indices to accurately estimate monthly vegetation cover (e.g. Multi Sensor Vegetation Index; Moradizadeh and Saradjian, 2016).

6.5 Monthly carbon inputs

6.5.1 C inputs under BAU practices

Carbon inputs for the BAU scenarios shall be estimated using the approach proposed by Smith *et al.* (2005; 2006; 2007) and Gottschalk *et al.* (2012). Total plant C inputs to the soil, which include plant litter, root exudates and fine root turnover, are rarely known. To overcome this problem, RothC shall be run in 'equilibrium mode' to calculate the initial plant carbon inputs to the soil (or 'equilibrium Carbon inputs', Ceq), which led to the initial SOC stocks (GSOCmap), under historic forcing conditions. The Ceq thus represents the historical average of annual carbon input of the BAU scenario up to the year 2000. For further details on the equilibrium run and initialization to estimate Ceq, refer to section 3.2 (General modeling procedures). Once these initial carbon inputs have been established (from the year 2000 onwards), year–to–year changes can be adjusted in accordance with changes in Net Primary Production (NPP), as changes in C inputs to the soil are assumed to be associated with changes in NPP (Smith *et al.*, 2005). Thus, annual C inputs for the BAU scenario can be adjusted as:

$$BAU_{Ct} = C_{t-1} \times (NPP_{t-1}) - 1 \times NPP_t \tag{6.4}$$

where BAU_{Ct} is the annual carbon input of a specific year t; C_{t-1} is the annual carbon input of the previous year; NPP_t is the net primary production of year t, and $NPP - t$ is the NPP of the previous year (in tC/ha). Thus, the average NPP over the initialization period shall be associated with Ceq and the annual C inputs for the BAU scenario can be adjusted as:

$$BAU_{Ct\ 2001} = C_{eq} \times NPP^{-1}_{1980-2000} \times NPP_{2001} \tag{6.5}$$

where $BAU_{Ct\ 2001}$ is the annual carbon input for the first year of the 'short spin–up' phase; C_{eq} is the estimated annual C input at equilibrium derived through the 'long spin–up' process; $NPP_{1980\text{˙}2000}$ is the estimated average net primary production over the initialization period (1980–2000); and NPP_{2001} is the estimated annual net primary production for the first year of the 'short spin–up' phase. The annual C inputs for the BAU scenario can be then adjusted following equation 7, according to changes in the NPP. The estimation of NPP using the MIAMI model (Lieth, 1975) is defined as the standard method in this document. It requires little input and is easily applicable worldwide, can be used to estimate NPP under future climatic conditions, and can act as a baseline

for different NPP data sets or projections (e.g. Gottschalk *et al.*, 2012). NPP estimated with the MIAMI model is computed with the following equations:

$$NPP_{MIAMI} = min(NPP_T, NPP_P) \tag{6.6}$$

$$NPPT_{MIAMI} = \frac{3000}{1 + e^{1.315 - 0.119} \times T} \tag{6.7}$$

$$NPPP_{MIAMI} = 3000 \times 1 - e^{-0.000664P} \tag{6.8}$$

where NPP is the climatic net primary production in dry matter (DM; $g/m^2/yr^1$), NPP_T is the temperature dependency term of NPP, where T is the annual mean temperature (°C) and NPP_P is the moisture dependency term of NPP, where P is the mean annual sum of precipitation (mm). NPP is limited by either temperature or precipitation. MIAMI model annual NPP can be expressed in tC/ha as:

$$NPP_{MIAMI}\ tCha^1yr^{-1} = NPP_{MIAMI}(DM; gm^{-2}yr^{-1}) \times 0.01 \times 0.48 \tag{6.9}$$

The annual NPPMIAMI shall be estimated for each grid cell from the climatic data sets described in section 6.1 for the different simulation periods (1981–1990; 1991–2000; 2001–2010; 2011–2020; 2021–2040). The NPPMIAMI is used to estimate BAU carbon inputs under current and projected climatic conditions. The change in NPP is used as a surrogate for estimating the change in C input and assumes that a similar proportion remains in the field (e.g. Smith *et al.*, 2005; Gottschalk *et al.*, 2012). In a first instance, countries should focus on C inputs in agricultural lands in 2020, the use of which has not changed since the year 2000. Changes in land use and management over the period 2000–2020 and associated changes in C inputs can nevertheless be taken into account, if trends in biomass removal are known, in order to adjust C inputs (e.g. Schulze *et al.*, 2010; Plutzar *et al.*, 2016; Neumann and Smith, 2018). Thus, the annual changes in C inputs by equations 7 and 8 can be adjusted using annual land cover data. For example, by assuming and approving an NPP of 12, 28 and 47 percent for forests, grasslands and croplands (Schulze *et al.*, 2010), the annual NPP of a specific year (NPPt) can be adjusted using these coefficients (equations

6.9 to 6.11), and the annual C inputs can then be estimated by equations 6.3 and 6.4:

$$NPPt_{forests} = NPP_{MIAMI} \times 0.88 \tag{6.10}$$

$$NPPt_{grasslands} = NPP_{MIAMI} \times 0.72 \tag{6.11}$$

$$NPPt_{croplands} = NPP_{MIAMI} \times 0.53 \tag{6.12}$$

6.5.2 C inputs under SSM practices

SSM practices shall be grouped into three scenarios as a standard method, based on their expected relative effects on C inputs compared to BAU: Low, Medium and High C inputs. The SSM practices considered in this approach are practices that affect C inputs to the soil, as changes in C inputs have been identified as one of the factors to which models are most sensitive when projecting changes in SOC stocks (FAO, 2019). As with estimates of BAU C inputs, total plant C inputs to the soil, including plant litter, root exudates and fine root turnover, are rarely known. Thus, C inputs of SSM scenarios will represent a percent increase from BAU C inputs:

$$\Delta\%CSSM - BAU = (Cinputs_{SSM} - Cinputs_{BAU}) \times Cinputs_{BAU} \tag{6.13}$$

As a standard, the expected effects (percent increase in C inputs) of 3 scenarios have been conservatively set at:

- Low: 5 percent increase in C inputs
- Medium: 10 percent increase C inputs
- High: 20 percent increase in C inputs

These percentages (based on Smith, 2004; Wiesmeier *et al.*, 2016) shall be used to produce the mandatory maps for the global product. An additional 'High

increase' scenario, considering a 30 percent increase in C inputs, can be modeled, to compare results with recent 'top–down' modeling approaches (e.g. CIRCASA). The use of default percentages in C input increase can be applied globally without complex configuration. However, countries should carefully check whether these scenarios are reasonable and under what type of management practices they are achievable. Countries are encouraged to produce and provide additional maps, taking into account their own estimates of the effects of different selected practices or land use changes, based on expert knowledge and local capacities. These effects can be determined on the basis of expert opinion and available information at the country level. A meta–analysis should be conducted based on the latest available local and regional studies to estimate how agricultural practices affect average annual C inputs (and the percent increase in C input compared to BAU practices). These practices may include, for example, the use of cover crops, rotation with high residue yielding crops or perennials, residue retention, grazing management, plant nutrition, species introduction, manure or organic amendment application, among others. If no data is directly provided in the compiled studies, carbon inputs and percent increase in C inputs relative to BAU practices shall be estimated considering the framework proposed by Bolinder *et al.* (2007). The annual C inputs required to model the effects of SSM practices under 3 scenarios (Low, Medium, High) for each modeling unit (i.e. grid cells) shall be estimated from the annual BAU C inputs:

$$SSM_{Ct}tCha^{-}1^{y}r^{-}1^{=}BAU_{Ct} + \%\Delta CSSM_i - BAU \times BAU_{Ct} \tag{6.14}$$

where SSM_{Ct} represents the estimated annual C inputs for a specific scenario (i =Low, Medium, High) for year t; BAU_{Ct} represents the estimated annual C inputs for the BAU scenario for year t (determined from C inputs at equilibrium, as explained at the beginning of this section and in Chapter 5), and $\%\Delta CSSM_i - BAU$ is the representative percent increase in C inputs for a specific scenario (i=Low, Medium, High).

6.6 Residue decomposability: DPM/RPM

Default values for the decomposability of incoming plant material reflected in the ratio of decomposable plant material (DPM) to resistant plant material (RPM)(DPM/RPM) can be used (e.g. 1.44 for crops and improved grasslands;

0.67 for unimproved grasslands and shrublands, and 0.25 for forests, woodlands and tree crops; Falloon and Smith, 2009). Table 6.1 (Land cover data sets) show default DPM/RPM for FAO land use classes. These default values can be modified according to region–specific data and local knowledge.

6.7 Required data sets and global data sources.

The required data sets described in this chapter are summarized here:

Global and regional data sources to generate national SOCseq maps.

Type	Source	Address	Resolution
Climatic monthly data	TerraClimate	https://developers.google.com/earth-engine/ datasets/catalog/ IDAHO_EPSCOR_TERRACLIMATE	4 x 4 km
SOC stocks 0-30 cm	GSOCmap - FAO-ITPS	http://54.229.242.119/GSOCmap/	1 x 1 km
Soil Texture	Harmonized World Soil Database v1.2	http://www.fao.org/soils-portal/	1 x 1 km
Soil Texture	OpenGeoHub Foundation	https://doi.org/10.5281/zenodo.1476854	250m
Soil texture, (including uncertainties)	Soil Grids -ISRIC	http://soilgrids.isric.org	250 m
NDVI- Historic images (2001-2020) every 16 days	MODIS	https://lpdaac.usgs.gov/ products/mod13a2v006/	1 x 1km
Land Cover	MODIS Land Cover Dynamics MCD12Q2	https://modis.gsfc.nasa.gov/ data/dataprod/mod12.php	500 x 500m 1 x 1 km
Land Cover	European Space Agency (ESA) Climate Change Initiative (CCI) Copernicus Climate Change Service (C3S)	https://www.esa-landcover-cci.org/	300 x 300m
Land Cover – Land Use	FAO Global Land Cover SHARE	http://www.fao.org/land-water/land/ land-governance/land-resources- planning-toolbox/category/details/en/c/1036355/	1 x 1km
Land Cover	USGS Global Land Survey	https://lta.cr.usgs.gov/GLS	30 x 30m
Land Cover	CORINE land cover(Europe only)	https://land.copernicus.eu/ pan-european/corine-land-cover	100 x 100 m

The proposed regional and global data sources to obtain the required input data when no quality national or regional data is available are described here:

Land cover aggregation schemes into RothC land use classes. Example from ESA.

ESA Land Cover Class	ESA class	RothC Land Use type
Cropland rainfed	10	Agricultural crops/ improved grassland
Cropland rainfed herbaceous cover	11	Agricultural crops/ improved grassland
Mosaic Cropland > 50%	30	Agricultural crops/ improved grassland
Cropland - Tree/shrub cover	12	Forest/Deciduous/ tropical woodland
Cropland irrigated flooding	20	Waterlogged soils
Grasslands	130	-9999
Mosaic Natural vegetation herbaceous >50% cropland	40	Unimproved grassland and scrub (including Savanna)
Mosaic herbaceous cover >50%/ trees-shrubs	110	Agricultural crops/ improved grassland
Shrubland	120	Unimproved grassland and scrub/Savanna
Shrubland evergreen	121	Unimproved grassland and scrub/Savanna
Shrubland deciduous	122	Unimproved grassland and scrub/Savanna
Tree cover broadleaved deciduous open 15-40%	62	Unimproved grassland and scrub/Savanna
Tree cover needle leaved deciduous open 15-40%	82	Unimproved grassland and scrub/Savanna
Tree cover broadleaved evergreen closed to open >15%	50	Unimproved grassland and scrub/Savanna
Tree cover broadleaved deciduous closed to open >15%	60	Forest/Deciduous/ tropical woodland
Tree cover broadleaved deciduous closed >40%	61	Forest/Deciduous/ tropical woodland
Tree cover needle leaved	70	Forest/Deciduous/

ESA Land Cover Class	ESA class	RothC Land Use type
evergreen closed to open >15%		tropical woodland
Tree cover needle leaved evergreen closed >40%	71	Forest/Deciduous/ tropical woodland
Tree cover needle leaved evergreen open >40%	72	Forest/Deciduous/ tropical woodland
Tree cover needle leaved deciduous closed to open >15%	80	Forest/Deciduous/ tropical woodland
Tree cover needle leaved deciduous closed >40%	81	Forest/Deciduous/ tropical woodland
Tree cover mixed leave type	90	Forest/Deciduous/ tropical woodland
Mosaic tree-shrub >50%/ herbaceous cover	100	Forest/Deciduous/ tropical woodland
Shrub or herbaceous flooded fresh/saline/brackish water	180	Forest/Deciduous/ tropical woodland
Tree cover flooded fresh or brackish water	160	Waterlogged
Tree cover flooded saline water	170	-9999
Urban areas	190	-9999
Lichens and mosses	140	-9999
Bare areas	200	Others - No data-9999
Sparse vegetation tree-shrub-herbaceous (<15%)	150	Others - No data-9999
Sparse tree (<15%)	151	Others - No data-9999
Sparse Shrub (<15%)	152	Others - No data-9999
Sparse herbaceous (<15%)	153	Others - No data-9999
Consolidated bare areas	201	Others - No data-9999
Unconsolidated bare areas	202	Others - No data-9999
Permanent snow/ice	220	Others - No data-9999
Water bodies	210	Others - No data-9999
No data	0	Others - No data-9999

cover classes= "-9999" denotes areas to be excluded without local adaptations in the RothC model.

Chapter 7

Software environment

A spatially explicit version of the RothC model (e.g. Gottschalk **et al.**, 2012; Mondini *et al.* 2017; Morais *et al.*; 2019) is required to generate national SOC sequestration maps. A spatialized version of the model was developed by the GSP Secretariat using an open source R–environment, based on the SoilR package developed by Sierra *et al.* (2012; downloadable from the Max Planck Institute of Biogeochemistry (https://www.bgc\T1\textendashjena.mpg.de/TEE/software/soilr/). Other spatialized versions have been developed in Fortran (e.g. Gottschalk *et al.*, 2012) and MATLAB (e.g. Morais *et al.*, 2019) environments. Users can use these and other alternative local adaptations of a spatialized RothC model by following the general procedures and input data described in Chapter 5 and 6 to obtain consistent results. Users are asked to provide evidence, i.e. .peer–reviewed scientific journal papers university theses etc.„ demonstrating that the use of the modified version and changes in model parameters are appropriate for the selected agro–ecological conditions. This Chapter summarizes the steps required to set–up the software environment (R, RStudio) to prepare the input data and run the spatialized R–version of the RothC model through scripts provided by the GSP. Additional supplementary scripts for QGIS and Google Earth Engine (GEE) are also provided. Users are required to download the following open source software:

- RStudio (latest stable version, https://rstudio.com/products/rstudio/download/)

- QGIS (latest stable version, https://qgis.org/en/site/forusers/download.html)

In order to use Google Earth Engine users must register an account. The instructions in this Chapter will guide users through installing and manually configuring the software to be used for Microsoft Windows desktop platform. Instructions for the other platforms (e.g. Linux Flavours, Mac OS) can be found through free online resources.

7.1 Setting up the R environment: R, RStudio, and R Packages

R is a language and environment for statistical computing. It provides a wide variety of statistical (e.g. linear modeling, statistical tests, time–series, classification, clustering, etc.) and graphical methods. It is highly extensible through packages, which are collections of R functions, data and documentation.

7.1.1 Obtaining and installing R

In order to run RStudio, R base needs to be installed according to the user's environment. Installation files and instructions can be downloaded from the Comprehensive R Archive Network (CRAN) following these steps:

1. Go to the following link https://cloud.r\T1\textendashproject.org/index.html to download and install R.
2. Pick an installation file for your platform.

7.1.2 Obtaining and installing RStudio

Since R base does not have a Graphical User Interface (GUI), first time users may encounter some difficulties in running the provided scripts. There are some GUIs which offer some of the functionality of R. RStudio makes R easier to use. It includes a code editor, debugging and visualization tools. Similar steps need to be followed to install RStudio:

1. Go to https://www.rstudio.com/products/rstudio/download/ to download and install RStudio's open source edition.
2. On the download page, RStudio Desktop, Open Source License option should be selected.
3. Pick an installation file for your platform.

7.1.3 Getting started with R

One of the main benefits of using R is that this programming language relies on a wide and active community of developers and end–users. Several manuals, including the one found on CRAN below, guiding material, tutorials and web pages dedicated to debugging errors, such as stackoverflow (listed below) can be found online:

- R manuals: http://cran.r-project.org/manuals.html
- Contributed documentation: http://cran.r-project.org/other-docs.html
- Quick–R: http://www.statmethods.net/index.html
- Stackoverflow R community: https://stackoverflow.com/questions/tagged/r

7.2 R packages

As mentioned previously, the main advantage of R is its extensibility. The scope of the possible implementations of R can be greatly increased with the vast collection of packages that extend its basic functionalities. Packages are the equivalent of add–ons that developers can freely write and make available through the open–source platform that constitutes R.

7.2.1 Finding R packages

The primary source for R packages is CRAN's official website, where currently about 12 000 available packages are listed. For spatial applications, various packages are available. You can obtain information about the available packages directly on CRAN with the available.packages() function. The function returns a matrix of details corresponding to the packages currently available at one or

more repositories. An easier way to browse the list of packages is using the Task Views link, which groups together packages related to a given topic. The following Subsections present some of the most used packages related to digital soil mapping.

7.2.2 Some useful R–packages for the SOCseq maps

R has a large and growing number of spatial data packages. We recommend taking a quick browse on R's official website to see the spatial packages available: http://cran.r-project.org/web/views/Spatial.html. Some of these packages that will be useful to generate the national SOCseq maps include:

- *raster*: Reading, writing, manipulating, analyzing and modeling of gridded spatial data. The package implements basic and high–level functions, processing of very large files is supported.
- *rgdal*: Provides bindings to the 'Geospatial' Data Abstraction Library ('GDAL') (>= 1.11.4) and access to projection/transformation operations from the 'PROJ' library. Use is made of classes defined in the 'sp' package. Raster and vector map data can be imported into R, and raster and vector 'sp' objects exported.
- *ncdf4*: Provides a high–level R interface to data files written using unidata's netCDF library (version 4 or earlier), which are binary data files that are portable across platforms and include metadata information in addition to the data sets. Using this package, netCDF files (either version 4 or "classic'' version 3) can be opened and datasets read in easily.
- *SoilR*: This package contains functions for modeling Soil Organic Matter decomposition in terrestrial ecosystems. See https://cran.r-project.org/web/packages/SoilR/SoilR.pdf.
- *abind*: Combine multidimensional arrays into a single array. This is a generalization of 'cbind' and 'rbind'. Works with vectors, matrices, and higher–dimensional arrays. Also provides functions 'adrop', 'asub', and 'afill' for manipulating, extracting and replacing data in arrays.
- *soilassessment*: Soil assessment builds information for improved decision in soil management. It analyzes soil conditions with regard to agriculture crop suitability requirements (such as those given by FAO http://www.fao.org/land-water/databases-and-software/crop-information/en/ soil fertility classes, soil erosion models and soil salinity classification. Suitability

requirements are for crops grouped into cereal crops, nuts, legumes, fruits, vegetables, industrial crops, and root crops.

7.2.3 Installation of R–packages used in this technical manual

The authors of this Technical Manual used a number of different R packages (summarized in Table 7.1). All required packages can be installed using the following code and the install.packages() function when starting a new SOC sequestration mapping project:

```
install.packages("name_of_the_package")
```

Alternatively, the code for the installation of the needed packages is included at the beginning of each Chapter.

Required R–packages for the national SOC sequestration potential maps using an R–environment.

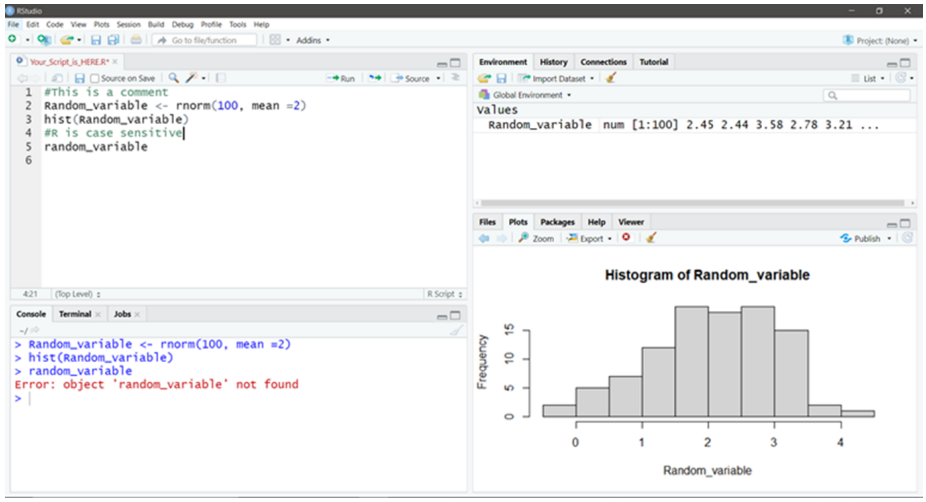
Protocol application area	R package	Reference
Import and export raster data	raster	Hijmans et col. (2020)
Import and export raster data	ncdf4	David Pierce (2019)
Import and export vector data	rgdal	Bivand et col (2019)
Harmonization	raster	Hijmans et col. (2020)
Harmonization	rgdal	Bivand et col (2019)
RothC model	SoilR	Sierra and Mueller (2014)
Data manipulation	abind	Plate (2016)
RothC model, NPP MIAMI model	soilassessment	Omuto (2020)

7.3 Considerations when using R

It is important to note the following points when using R:

- As shown in Figure 7.1, R is a case–sensitive scripting software. More than 90 percent of its commands are scripted in a text–editor and executed by running the line/script.

- Hash (#) denotes the beginning of a comment and is not executed by the software. Consequently, it can be used to insert comments in a line. All comments after hash (#) are colored green (like green traffic light) implying "pass" without execution.
- With the default editor theme errors and warnings are given in red, while functions and numbers are given in blue and commands and variables are given in black.
- When using RStudio text–editor, four panes are available in which the top left pane is the text editing window, top right pane is for the data environment, bottom right pane is for display and help, and the bottom left is the console for executing the scripts.
- Implemented scripts and reports (warning or errors) are shown in the console panel.
- Some commands may run for some time and patience is recommended to enable the software to progress to completion. During such time, a red icon will be shown at the top left corner of the console panel.
- The execution of a line or script can be stopped if necessary by clicking on the "STOP" button from the tools bar.



Overview of RStudio's windowpanes and functionalities

Chapter 8

Running the scripts: overview

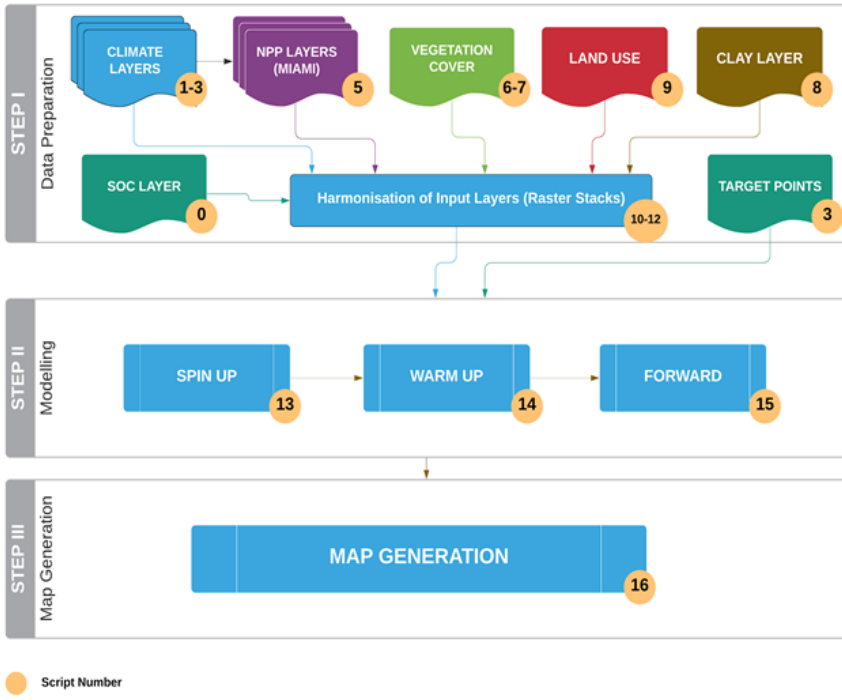
8.1 Summary of steps and scripts

The generation of national SOC sequestration potential maps using the spatially explicit R–version of the RothC model shall be divided in three stages (Figure 8.1.):

1. Preparation and Harmonization of data (consists of eleven R scripts, one QGIS model script, and one Google Earth Engine script)
2. Running the model in three phases (three R scripts),
3. Transformation of vector data to raster data (map generation, one R script).

8.2 Stage 1: Preparation of data

Running the model over an area will require several spatial layers of information (climate, clay content, land use, vegetation cover, NPP layers) and defining target points where the model will be run. So before running the model, we will "harmonize" the different spatial layers, in order to have the same extent, same pixel size and same Coordinate Reference System (CRS). On the other



Workflow for generating the national GSOCseq layers.

hand, each modeling phase (spin up, short spin–up/warm up, forward runs) will require a different selection of layers for the different time series. So we will create 'stacks' of the different layers for the single modeling phases. Land cover classes need to be re–classified into land use types that the model will recognize. We will reclassify land cover classes (like the ones provided by ESA; European Space Agency) to match the FAO land cover classes. At this stage, we will also create other input layers (like NPP and vegetation cover layers) which are necessary to run the model. Finally, we will create target points over the land use classes of interest (agricultural lands). These target points will become the modeling units (where the model is to be run). So, the first step (Fig. 8.1; see Chapter 9) is aimed at:

- preparing, organizing and harmonizing all the required input data layers to run the model in the different phases
- creating supplementary input data layers
- creating target points for land use classes of interests

Eleven R scripts, one QGIS script and one Google earth engine script are provided to complete these tasks (Table 8.1.).

8.3 Stage 2: Running the model

Once the input data layers are prepared and stacked, we will run the spatialized RothC model at each target point using three specific scripts (Table 8.1. in the following three phases (Chapter 10):

1. Long spin up phase: he equilibrium carbon inputs (annual carbon inputs in t/ha required to reach SOC stocks in year 2000) and the initial stocks of the different SOC pools are calculated;
2. Short spin up or Warm Up phase: SOC stocks are adjusted for the 2000–2020 period;
3. Forward" phase: SOC stocks are projected (2020 to 2040) under a "business as usual" scenario (no changes in carbon inputs), "low" scenario (5 percent increase in carbon inputs), "medium" scenario (10 percent increase in carbon inputs), and "high" scenario (20 percent increase in carbon inputs).

8.4 Stage 3: generation of the map

After the "forward" modeling phase, in the final step we will calculate the average absolute and relative SOC sequestration rates over a 20–years period for each scenario and for each target point. The vector target points will be then rasterized and saved to geotiff format to obtain the final product, using a specific R script. All the provided scripts are summarized in Table 8.1.

Summary of the scripts for the complete modelling process.

Type of Layer	Script	Objective
SOC layer	R- Script number 0	Cut the SOC layer by the area of interest polygon
Climate layers	R- Script number 1	Rearrangement of climate layers
	R- Script number 2	(TerraClimate layers from .ncd to .tif)
	R- Script number 3	
NPP layers	R- Script number 5	Creation of NPP layers
Vegetation Cover (VC)	GEE Script number (Google Earth Engine)	Creation of VC layers
	R- Script number 7	
Clay layers	R-Script number 8	Obtaining clay contents 0-30 cm from different depths (ISRIC)
Land Use layer	R-Script number 9	Re-classification into FAO land cover classes
STACK for SPIN UP	R-Script number 10	Stack input data layers for the spin up phase
STACK for WARM UP	R-Script number 11	Stack input data layers for the warm up phase
STACK for FORWARD	R-Script number 12	Stack input data layers for the forward phase
Target points	QGis model script	Creation of target points
SPIN UP	R- Script number 13	Run long spin up phase
WARM UP	R- Script number 14	Run warm up phase
FORWARD	R- Script number 15	Run forward phase
POINTS TO RASTER	R- Script number 16	Rasterize points

Chapter 9

Stage 1: preparation of input data

This stage is aimed at:

- preparing, organizing and harmonizing all the required input data layers to run the model in the different phases;
- creating supplementary input data layers;
- creating target points for land use classes of interests.

During this stage we will need to arrange and prepare climate datasets for the different modelling phases, generate NPP estimates for each phase, generate vegetation cover data, prepare clay content data layers, and harmonize and stack all layers for each modelling phase. Finally, we will have to create target points to run the model. This stage requires the most effort and is the most time consuming of the entire process. Eleven R scripts, one QGIS script and one Google earth engine script are provided to complete these tasks.

9.1 Preparation of SOC layer

As a default option, users are invited to use the GSOCmap to retrieve their SOC data for their area of interest (AOI). This can be achieved easily, by clipping

SOC layer AOI

Input	Output
FAO SOC map: GSOCmap_v1.5.0.tif Area of interest (AOI): COUNTRY_POLYGON.shp	SOC map of AOI SOC_MAP_[country_code].tiff

the GSOCmap to the extent of a shapefile making up the borders of the chosen study area or country. All data sources can be found in Table 6.3 of Chapter 6.

9.1.1 Script Number 0: "SOC_MAP_AOI.R". Preparation of the SOC layer. Inputs and Outputs

First, open the script SOC_MAP_AOI.R in RStudio. If you haven't done so previously, install the necessary packages. Then create two user-defined variables containing the paths to the two working directories: * "WD_AOI" which contains the vector polygon of the AOI; * "WD_GSOC", which contains the GSOCmap raster layer

```
#Install all necessary packages
install.packages(c("raster","rgdal","SoilR",
                   "Formula","soilassessment","abind","ncdf4",
                   "foreach","doParallel"))

#Load the packages into R
library(raster)
library(rgdal)

# Set the path to GSOCmap and Area of interest (AOI) vector.
WD_AOI<-("C:/TRAINING_MATERIAL/INPUTS/AOI_POLYGON")
WD_GSOC<-("C:/TRAINING_MATERIAL/INPUTS/SOC_MAP")

# Open the shapefile of the AOI (region/country)
setwd(WD_AOI)
AOI<-readOGR("Departamento_Pergamino.shp")
```

```
#Open FAO GSOC MAP
setwd(WD_GSOC)
SOC_MAP <- raster("GSOCmap_1.6.1.tif")
```

Finally, we clip the SOC layer with the vector polygon of the AOI and save the result to the WD_SOC folder. This layer will become the master layer of the process.

```
SOC_MAP_AOI<-crop(SOC_MAP,AOI)
SOC_MAP_AOI<-mask(SOC_MAP_AOI,AOI)
writeRaster(SOC_MAP_AOI,filename="SOC_MAP_AOI.tif",
            format="GTiff", overwrite = TRUE)
```

9.2 Preparation of climate Layers (based on TerraClimate)

The climate variables needed for the three modeling phases are:

1. Monthly rainfall (mm/month);
2. Monthly Evapotranspiration (mm/month);
3. Average monthly mean air temperature (average °C/month).

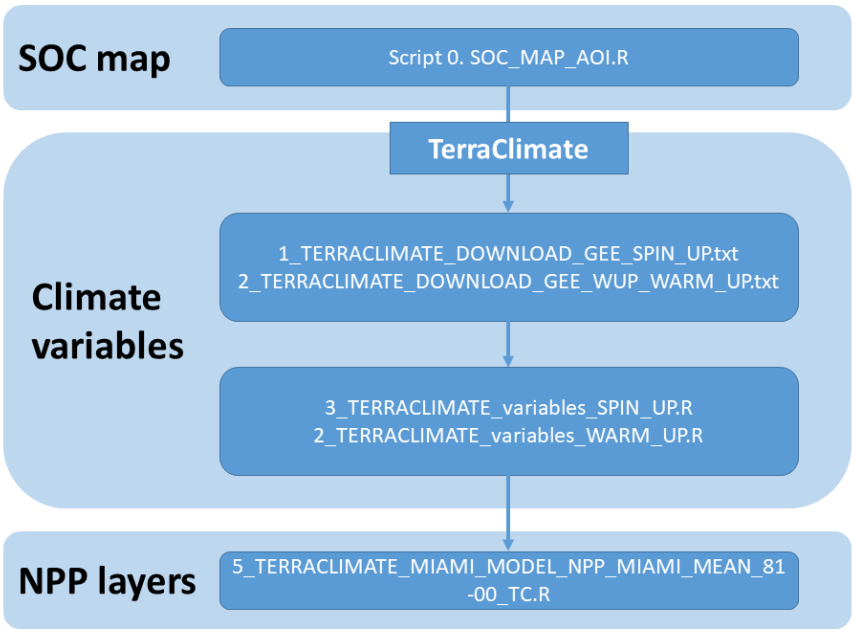
We will need to arrange these climatic variables into three datasets:

- 1980-2000 (monthly average values for the complete series)
- 2001-2020 (year to year monthly values)
- 2001-2020 (monthly average values for the complete series)

Gridded climate data shall be obtained from either National Sources or regional or global datasets when national gridded historical climate datasets are not available. The recommended global data source of these layers is TerraClimate (readily available from the Google Earth Engine catalog: https://developers.google.com/earth-engine/datasets/catalog/IDAHO_EPSCOR_TERRACLIMATE#citations)

For countries that want to use the TerraClimate dataset, several scripts will be presented on how to obtain and reformat the climate spatial layers, and to run the three modelling phases. Users can prepare the necessary input climate datasets using other data sources. However, these scripts may still be helpful to guide the preparation process of other datasets, and as a guide of the required outputs that will be needed as inputs for the different modeling phases.

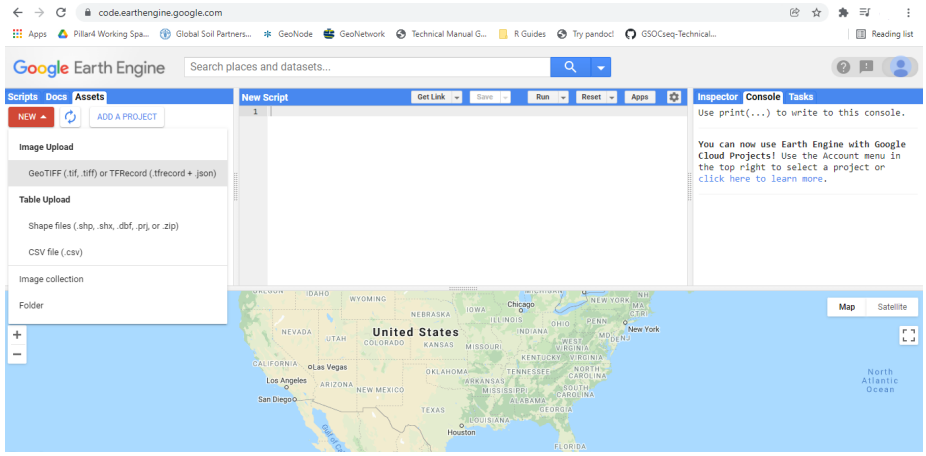
The preparation of the climate data with the TerraClimate dataset is presented in the flowchart below (Figure 9.0). To make use of the TerraClimate dataset, users need to first download the data for the time periods 1980-2000 and 2001-2018 using two scripts for Google Earth Egine (GEE) and subsequently prepare the target climatic variables using two R scripts.



Script order to follow for TerraClimate data sets.

9.2.1 Script Number 1. TerraClimate GEE Spin up phase

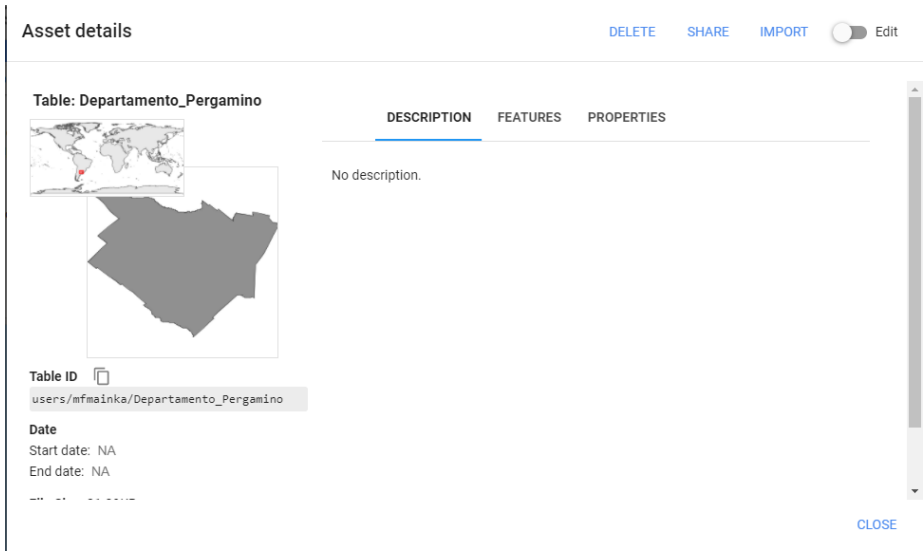
The TerraClimate dataset can be downloaded directly from Google Earth Engine, a powerful and free cloud computing platform. First, the user will need to activate a Google Earth Engine account. To run the Google Earth Engine (GEE) tool, the user will need to copy and paste the script (provided below) into the GEE code editor (central panel, Fig. 9.1).



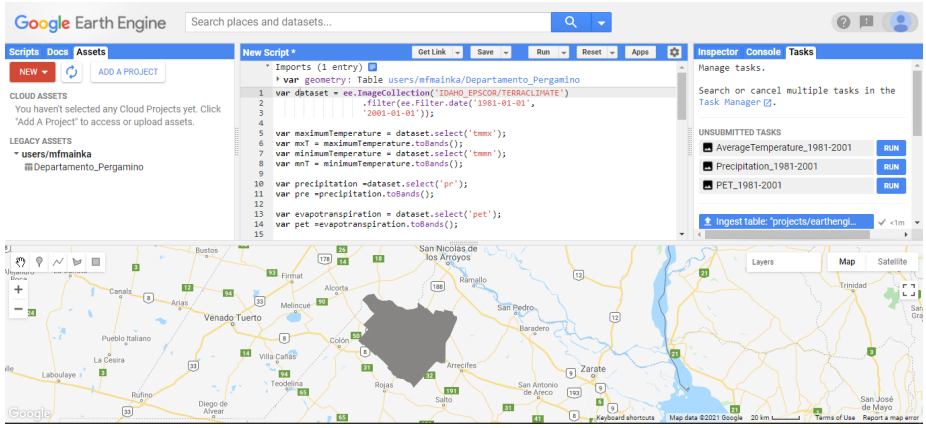
Google Earth Engine code editor

To run the script the user can input a new geometry using the geometry tools panel (Figure 9.2) or by uploading a shapefile (.shp) together with the other dependent files. To import the file, it is necessary to click on 'IMPORT' in the top right corner of the window. Then, we move to the code editor.

In the code editor (central panel), we see at the very top the section 'Imports (1 entry)'. In the line below we have to modify the name of our shapefile from 'polygon' to 'geometry'. This is necessary to run the following code without modifications.



Upload a shapefile of the Area of Interest (AOI)



After copying and pasting the following script in GEE, we run the script. The outputs are shown in the right panel under 'Tasks'. In the section 'Unbsubmitted tasks', we click on 'RUN' to export our generated files into a selected google drive folder (see Figure 9.3). The following script can be copied and pasted into GEE without any further modifications:

```
// Climate data sets for the Spin Up phase 1980-2001
// calculate the average temperature
// from minimum and maximum temperatures
// download the Minimum and Maximum Temperature
// download the Average temperature
// download the PET
// download the Precipitation

var dataset = ee.ImageCollection('IDAHO_EPSCOR/TERRACLIMATE')
                  .filter(ee.Filter.date('1981-01-01',
                  '2001-01-01'));

var maximumTemperature = dataset.select('tmmx');
var mxT = maximumTemperature.toBands();
var minimumTemperature = dataset.select('tmmn');
var mnT = minimumTemperature.toBands();

var precipitation =dataset.select('pr');
var pre =precipitation.toBands();

var evapotranspiration = dataset.select('pet');
var pet =evapotranspiration.toBands();

var diff = mxT.add(mnT);
var avT = diff.divide(2);

var avT =avT.clip(geometry);
var pre =pre.clip(geometry);
var pet =pet.clip(geometry);



Map.addLayer(avT, {}, 'default RGB');
Map.addLayer(pre, {}, 'default RGB');
Map.addLayer(pet, {}, 'default RGB');


var regionJSON = JSON.stringify(avT.getInfo());
```

```
Export.image.toDrive({
          image: avT,
          folder: "TerraClimate",
          description: 'AverageTemperature_1981-2001',
          scale: 4000,
          region: geometry
});

var regionJSON = JSON.stringify(pre.getInfo());
Export.image.toDrive({
          image: pre,
          folder: "TerraClimate",
          description: 'Precipitation_1981-2001',
          scale: 4000,
          region: geometry
});

var regionJSON = JSON.stringify(pet.getInfo());
Export.image.toDrive({
          image: pet,
          folder: "TerraClimate",
          description: 'PET_1981-2001',
          scale: 4000,
          region: geometry
});
```

9.2.2 Script Number 2. TerraClimate GEE Warm up and Forward phase

To retrieve the necessary climatic data (2001-2018/20) to be used as input for the warm up and subsequent forward phase the same steps are repeated as for Script 2.1. After defining a geometry or inputting a shapefile the following code can be copied and pasted into the GEE code editor.

```
// Climate data sets for the Warm Up phase and
// Forward phase 2001 - 2020
// calculate the average temperature
```

```
// from minimum and maximum temperatures
// download the Average temperature
// download the PET
// download the Precipitation

var dataset = ee.ImageCollection('IDAHO_EPSCOR/TERRACLIMATE')
        .filter(ee.Filter.date('2001-01-01', '2020-01-01'));

var maximumTemperature = dataset.select('tmmx');
var mxT = maximumTemperature.toBands();
var minimumTemperature = dataset.select('tmmn');
var mnT = minimumTemperature.toBands();

var precipitation =dataset.select('pr');
var pre =precipitation.toBands();

var evapotranspiration = dataset.select('pet');
var pet =evapotranspiration.toBands();

var diff = mxT.add(mnT);
var avT = diff.divide(2);

var avT =avT.clip(geometry);
var pre =pre.clip(geometry);
var pet =pet.clip(geometry);



Map.addLayer(avT, {}, 'default RGB');
Map.addLayer(pre, {}, 'default RGB');
Map.addLayer(pet, {}, 'default RGB');



var regionJSON = JSON.stringify(avT.getInfo());
Export.image.toDrive({
          image: avT,
          folder: "TerraClime",
```

```
          description: 'AverageTemperature_2001-2021',
          scale: 4000,
          region: geometry
});

var regionJSON = JSON.stringify(pre.getInfo());
Export.image.toDrive({
          image: pre,
          folder: "TerraClime",
          description: 'Precipitation_2001-2021',
          scale: 4000,
          region: geometry
});

var regionJSON = JSON.stringify(pet.getInfo());
Export.image.toDrive({
          image: pet,
          folder: "TerraClime",
          description: 'PET_2001-2021',
          scale: 4000,
          region: geometry
});
```

9.2.3 Script Number 3. TerraClimate Variables Spin up phase

Once the data has been downloaded using GEE for the time period 1981-2000, the necessary target variables for the spin up phase can be prepared using the following scripts. For each modelling phase we will need a different selection of climate layers. For phase 1 ("Long Spin up"), we will need to stack 12 spatial layers (the output file will be a multiband raster layer) for each climate variable mentioned above (temperature, precipitation and evapotranspiration). The time series for this initial phase goes from 1981 to 2000. The script number 3 will transform the downloaded TerraClimate files to obtain monthly averages (temperature, precipitation, evapotranspiration) for the 1981-2000 series, ready to be used in the spin up modelling phase.

```
#DATE:  2/11/2020

#  MSc Ing Agr Luciano E. Di Paolo
#  Dr Ing Agr Guillermo E Peralta
##########################################

library(raster)
library(rgdal)


# TerraClimate FROM GOOGLE EARTH ENGINE
#Abatzoglou, J.T., S.Z. Dobrowski, S.A. Parks,
# K.C. Hegewisch, 2018, Terraclimate,
#a high-resolution global dataset of monthly climate
#and climatic water balance from 1958-2015, Scientific Data,
##########################################

#Set working directory
WD<-("C:/TRAINING_MATERIAL/INPUTS/TERRA_CLIMATE")
setwd(WD)


# Open the TerraClimate data from GEE

tmp<-stack("AverageTemperature_1981-2001.tif")

pre_81_00<-stack("Precipitation_1981-2001.tif")

pet_81_00<-stack("PET_1981-2001.tif")


# TEMPERATURE

# Get one month temperature ( January)

tmp_Jan_1<-tmp[[1]]

dim(tmp_Jan_1)
```

```
# Create empty list

Rlist<-list()

# Average of 20 years (j)  and 12 months (i)

######for loop starts#######
for (i in 1:12) {

var_sum<-tmp_Jan_1*0
k<-i

for (j in 1:20) {
print(k)
var_sum<-(var_sum + tmp[[k]])

k<-k+12
}

#Calculate each month average.

var_avg<-var_sum/20

# Save the average of each month (i)

Rlist[[i]]<-var_avg
}
#######for loop ends########

#save a stack of months averages

Temp_Stack<-stack(Rlist)
Temp_Stack<-Temp_Stack*0.1 # rescale to C
writeRaster(Temp_Stack,
filename='Temp_Stack_81-00_TC.tif',"GTiff",overwrite=TRUE)

#PRECIPITATION
```

```
# Get one month Precipitation ( January)

pre_Jan_1<-pre_81_00[[1]]

dim(pre_Jan_1)

# Create empty list

Rlist<-list()

# Average of 20 years (j)  and 12 months (i)

######for loop starts#######
for (i in 1:12) {

var_sum<-pre_Jan_1*0
k<-i

for (j in 1:20) {
print(k)
var_sum<-(var_sum + pre_81_00[[k]])
k<-k+12
}
#Save each month average.

var_avg<-var_sum/20

Rlist[[i]]<-var_avg
}

######for loop ends#######

#save a stack of months averages

Prec_Stack<-stack(Rlist)
writeRaster(Prec_Stack,
filename='Prec_Stack_81-00_TC.tif',"GTiff",overwrite=TRUE)
```

```
# POTENTIAL EVAPOTRANSPIRATION


# Get one month PET ( January)

pet_Jan_1<-pet_81_00[[1]]

dim(pet_Jan_1)

# Create empty list

Rlist<-list()

# Average of 20 years (j)  and 12 months (i)

######for loop starts#######
for (i in 1:12) {

var_sum<-pet_Jan_1*0
k<-i

for (j in 1:20) {
print(k)
var_sum<-(var_sum + pet_81_00[[k]])

k<-k+12

}
#Save each month average.

var_avg<-var_sum/20


Rlist[[i]]<-var_avg
}
######for loop ends#######

#save a stack of months averages
```

```
PET_Stack<-stack(Rlist)
PET_Stack<-PET_Stack*0.1
writeRaster(PET_Stack,
filename='PET_Stack_81-00_TC.tif',"GTiff",overwrite=TRUE)
```

9.2.4 Script Number 4. TerraClimate variables warm-up phase

Once the data has been downloaded using GEE for the time period 2001-2018/20, the necessary target variables for the warm up and forward phases can be prepared using the following script. The purpose of the "Warm up" phase is to adjust the initial SOC stock and initial pools for the "forward" phase. Once the input climate layers have been harmonized, the model will run for each year from 2001 to 2018/20, using the monthly climate data of each year of the series (for 216/240 values for each month of the time series). The script number 4 is prepared to arrange the necessary TerraClimate files for this phase. We will need to generate one raster stack of 216/240 spatial layers for each climate variable mentioned above (216 spatial layers if we use just 18 years period instead of a 20 year period; from 2001 to 2018, depending on the available climate data). Each stack will have one layer for each month from 2001 to 2018/2020. For phase number 3, the "Forward" phase, we will need monthly averages of the time series 2001-2018/20. We will use the same arrangement as used in phase number one (one stack of 12 bands for each variable) but instead of using the averages of the 1981-2000 period we will use the climatic data of the 2001-2018/20 period. We will assume that there is no climate change in the next 20 years. Thus, script number 2 will also prepare the climate files for the "forward phase".

```
#DATE: 12/02/2021

#  MSc Ing Agr Luciano E. Di Paolo
#  Dr Ing Agr Guillermo E Peralta

# TerraClimate FROM GOOGLE EARTH ENGINE
#Abatzoglou, J.T., S.Z. Dobrowski, S.A. Parks,
# K.C. Hegewisch, 2018, Terraclimate,
#a high-resolution global dataset of monthly climate
# and climatic water balance from 1958-2015, Scientific Data,
```

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

##################

library(raster)
library(rgdal)

##################

WD<-(
"C:/TRAINING_MATERIALS/INPUTS/TERRA_CLIMATE")
setwd(WD)

# OPEN LAYERS


# Open the TerraClimate data from GEE

tmp<-stack("AverageTemperature_2001-2018.tif")

pre_01_18<-stack("Precipitation_2001-2018.tif")

pet_01_18<-stack("PET_2001-2018.tif")

# TEMPERATURE


# Get one month temperature ( January)

tmp_Jan_1<-tmp[[1]]

dim(tmp_Jan_1)

# Create empty list
Rlist<-list()

# Average of 20 years (j)  and 12 months (i)
##########for loop starts###############
```

```
for (i in 1:12) {
var_sum<-tmp_Jan_1*0
k<-i

for (j in 1:(dim(tmp)[3]/12)) {
print(k)
var_sum<-(var_sum + tmp[[k]])

k<-k+12

}
#Save each month average.

var_avg<-var_sum/(dim(tmp)[3]/12)

#writeRaster(ra,filename=name, format="GTiff")
Rlist[[i]]<-var_avg
}
###########for loop ends##############
#save a stack of months averages

Temp_Stack<-stack(Rlist)
Temp_Stack<-Temp_Stack*0.1 # rescale to C
writeRaster(Temp_Stack,
    filename='Temp_Stack_01-19_TC.tif',"GTiff",overwrite=TRUE)


#PRECIPITATION

# Have one month Precipitation ( January)

pre_Jan_1<-pre_01_18[[1]]

dim(pre_Jan_1)

# Create empty list
Rlist<-list()
```

```
# Average of 20 years (j)  and 12 months (i)

#########for loop starts############
for (i in 1:12) {

var_sum<-pre_Jan_1*0
k<-i

for (j in 1:(dim(pre_01_18)[3]/12)) {
print(k)
var_sum<-(var_sum + pre_01_18[[k]])

k<-k+12

}
#Save each month average.

var_avg<-var_sum/(dim(pre_01_18)[3]/12)

#writeRaster(ra,filename=name, format="GTiff",overwrite=TRUE)
Rlist[[i]]<-var_avg
}
###########for loop ends##########

#save a stack of months averages

Prec_Stack<-stack(Rlist)
writeRaster(Prec_Stack,
    filename='Prec_Stack_01-19_TC.tif',"GTiff",overwrite=TRUE)


# POTENTIAL EVAPOTRANSPIRATION

# Have one month ETP ( January)

pet_Jan_1<-pet_01_18[[1]]

dim(pet_Jan_1)
```

```
# Create empty list
Rlist<-list()

# Average of 18 years (j)  and 12 months (i)
#############for loop starts###############
for (i in 1:12) {

var_sum<-pet_Jan_1*0
k<-i

for (j in 1:(dim(pet_01_18)[3]/12)) {
print(k)
var_sum<-(var_sum + pet_01_18[[k]])

k<-k+12

}
#Save each month average.

var_avg<-var_sum/(dim(pet_01_18)[3]/12)

#writeRaster(ra,filename=name, format="GTiff",overwrite=TRUE)
Rlist[[i]]<-var_avg
}
##########for loop ends#############

#save a stack of months averages

PET_Stack<-stack(Rlist)
PET_Stack<-PET_Stack*0.1
writeRaster(PET_Stack,
  filename='PET_Stack_01-19_TC.tif',"GTiff",overwrite=TRUE)
```

9.2.5 Script Number 5. TerraClimate MIAMI model NPP mean 1981-2000

To adjust yearly C inputs during the warm up phase according to annual NPP values, we will need to estimate an average annual NPP 1981-2000, that will be used as the starting point to adjust C inputs during the "warm up" phase (See chapter 6). Script number 5 uses the TerraClimate climate raster outputs from script number 3 and estimates an annual NPP mean 1981-2000 value.

```
#DATE: 2-12-2020

# MSc Ing.Agr. Luciano E. DI Paolo
# PHD Ing.Agr. Guillermo E. Peralta


# MIAMI MODEL

library(raster)
library(rgdal)

WD_NPP<-("C:/TRAINING_MATERIALS/INPUTS/NPP")

WD_AOI<-(
  "C:/TRAINING_MATERIALS/INPUTS/AOI_POLYGON")

WD_GSOC<-(
  "C:/TRAINING_MATERIALS/INPUTS/SOC_MAP")

WD_TC_LAYERS<-(
  "C:/TRAINING_MATERIALS/INPUTS/TERRA_CLIMATE")

setwd(WD_TC_LAYERS)

# Open Annual Precipitation (mm) and
# Mean Annual Temperature (grades C) stacks

Temp<-stack("AverageTemperature_1981-2001.tif")
Prec<-stack("Precipitation_1981-2001.tif")
```

```
setwd(WD_AOI)
AOI<-readOGR("Departamento_Pergamino.shp")

#Temp<-crop(Temp,AOI)
#Prec<-crop(Prec,AOI)

# Temperature Annual Mean

k<-1
TempList<-list()
#######loop for starts#########
for (i in 1:(dim(Temp)[3]/12)){

Temp1<-mean(Temp[[k:(k+11)]])
TempList[i]<-Temp1

k<-k+12
}
#######loop for ends##########
TempStack<-stack(TempList)
TempStack<-TempStack*0.1 # rescale to C

#Annual Precipitation

k<-1
PrecList<-list()
########loop for starts#######
for (i in 1:20){

Prec1<-sum(Prec[[k:(k+11)]])
PrecList[i]<-Prec1

k<-k+12
}
########loop for ends#######
PrecStack<-stack(PrecList)

# Calculate eq 1 from MIAMI MODEL (g DM/m2/day)
```

```
NPP_Prec<-3000*(1-exp(-0.000664*PrecStack))

# Calculate eq 2 from MIAMI MODEL (g DM/m2/day)

NPP_temp<-3000/(1+exp(1.315-0.119*TempStack))

# Calculate eq 3 from MIAMI MODEL (g DM/m2/day)

NPP_MIAMI_List<-list()

########loop for starts#######
for (i in 1:20){
NPP_MIAMI_List[i]<-min(NPP_Prec[[i]],NPP_temp[[i]])
}
########loop for ends#######
NPP_MIAMI<-stack(NPP_MIAMI_List)

#NPP_MIAMI gDM/m2/year To tn DM/ha/year
NPP_MIAMI_tnDM_Ha_Year<-NPP_MIAMI*(1/100)

#NPP_MIAMI tn DM/ha/year To tn C/ha/year
NPP_MIAMI_tnC_Ha_Year<-NPP_MIAMI_tnDM_Ha_Year*0.5

# Save WORLD NPP MIAMI MODEL tnC/ha/year
setwd(WD_NPP)
writeRaster(NPP_MIAMI_tnC_Ha_Year,
  filename="NPP_MIAMI_tnC_Ha_Year_STACK_81-00.tif",
    format="GTiff",overwrite=TRUE)

#NPP_MIAMI_tnC_Ha_Year<-stack("NPP_MIAMI_tnC_Ha
# _Year_STACK_81-00.tif")

# NPP MEAN
NPP_MIAMI_MEAN_81_00<-mean(NPP_MIAMI_tnC_Ha_Year)


#Open FAO GSOC MAP
setwd(WD_GSOC)
```

```
SOC_MAP_AOI<-raster("SOC_MAP_AOI.tif")

# Crop & mask
setwd(WD_NPP)

NPP_MIAMI_MEAN_81_00_AOI<-crop(NPP_MIAMI_MEAN_81_00,AOI)
NPP_MIAMI_MEAN_81_00_AOI<-resample(NPP_MIAMI_MEAN_81_00_AOI,
                                   SOC_MAP_AOI)
NPP_MIAMI_MEAN_81_00_AOI<-mask(NPP_MIAMI_MEAN_81_00_AOI,AOI)

writeRaster(NPP_MIAMI_MEAN_81_00_AOI,
    filename="NPP_MIAMI_MEAN_81-00_AOI.tif",
      format="GTiff",overwrite=TRUE)
writeRaster(NPP_MIAMI_MEAN_81_00,
    filename="NPP_MIAMI_MEAN_81-00.tif",
      format="GTiff",overwrite=TRUE)


#UNCERTAINTIES MINIMUM TEMP , PREC
Temp_min<-Temp*1.02
Prec_min<-Prec*0.95

# Temperature Annual Mean

k<-1
TempList<-list()
########loop for starts#######
for (i in 1:20){

Temp1<-mean(Temp_min[[k:(k+11)]])
TempList[i]<-Temp1

k<-k+12
}
########loop for ends#######

TempStack<-stack(TempList)
TempStack<-TempStack*0.1 # rescale to C
```

```
#Annual Precipitation

k<-1
PrecList<-list()

########loop for starts#######
for (i in 1:20){

Prec1<-sum(Prec_min[[k:(k+11)]])
PrecList[i]<-Prec1

k<-k+12
}
########loop for ends#######

PrecStack<-stack(PrecList)

# Calculate eq 1 from MIAMI MODEL (g DM/m2/day)

NPP_Prec<-3000*(1-exp(-0.000664*PrecStack))

# Calculate eq 2 from MIAMI MODEL (g DM/m2/day)

NPP_temp<-3000/(1+exp(1.315-0.119*TempStack))

# Calculate eq 3 from MIAMI MODEL (g DM/m2/day)

NPP_MIAMI_List<-list()

########loop for starts#######
for (i in 1:20){
NPP_MIAMI_List[i]<-min(NPP_Prec[[i]],NPP_temp[[i]])
}
########loop for ends#######

NPP_MIAMI<-stack(NPP_MIAMI_List)

#NPP_MIAMI gDM/m2/year To tn DM/ha/year
```

```
NPP_MIAMI_tnDM_Ha_Year<-NPP_MIAMI*(1/100)

#NPP_MIAMI tn DM/ha/year To tn C/ha/year

NPP_MIAMI_tnC_Ha_Year<-NPP_MIAMI_tnDM_Ha_Year*0.5

# Save WORLD NPP MIAMI MODEL tnC/ha/year

setwd(WD_NPP)

writeRaster(NPP_MIAMI_tnC_Ha_Year,
   filename="NPP_MIAMI_tnC_Ha_Year_STACK_81-00_MIN.tif",
   format="GTiff",overwrite=TRUE)

# NPP MEAN

NPP_MIAMI_MEAN_81_00<-mean(NPP_MIAMI_tnC_Ha_Year)

# Crop & and mask

setwd(WD_NPP)

NPP_MIAMI_MEAN_81_00_AOI<-crop(NPP_MIAMI_MEAN_81_00,AOI)
NPP_MIAMI_MEAN_81_00_AOI<-resample(NPP_MIAMI_MEAN_81_00_AOI,
                                   SOC_MAP_AOI)
NPP_MIAMI_MEAN_81_00_AOI<-mask(NPP_MIAMI_MEAN_81_00_AOI,AOI)

writeRaster(NPP_MIAMI_MEAN_81_00_AOI,
  filename="NPP_MIAMI_MEAN_81-00_AOI_MIN.tif",
   format="GTiff",overwrite=TRUE)
writeRaster(NPP_MIAMI_MEAN_81_00,
  filename="NPP_MIAMI_MEAN_81-00_MIN.tif",
   format="GTiff",overwrite=TRUE)

#UNCERTAINTIES MAXIMUM TEMP , PREC

# Open Annual Precipitation (mm) and
# Mean Annual Temperature (C) stacks
```

```
Temp_max<-Temp*0.98
Prec_max<-Prec*1.05

# Temperature Annual Mean

k<-1
TempList<-list()

########loop for starts#######
for (i in 1:20){

Temp1<-mean(Temp_max[[k:(k+11)]])
TempList[i]<-Temp1

k<-k+12
}
########loop for ends#######

TempStack<-stack(TempList)
TempStack<-TempStack*0.1 # rescale to C

#Annual Precipitation

k<-1
PrecList<-list()

########loop for starts#######
for (i in 1:20){

Prec1<-sum(Prec_max[[k:(k+11)]])
PrecList[i]<-Prec1

k<-k+12
}
########loop for ends#######

PrecStack<-stack(PrecList)
```

```
# Calculate eq 1 from MIAMI MODEL (g DM/m2/day)

NPP_Prec<-3000*(1-exp(-0.000664*PrecStack))

# Calculate eq 2 from MIAMI MODEL (g DM/m2/day)

NPP_temp<-3000/(1+exp(1.315-0.119*TempStack))

# Calculate eq 3 from MIAMI MODEL (g DM/m2/day)

NPP_MIAMI_List<-list()

########loop for starts#######
for (i in 1:20){
NPP_MIAMI_List[i]<-min(NPP_Prec[[i]],NPP_temp[[i]])
}
########loop for ends#######

NPP_MIAMI<-stack(NPP_MIAMI_List)

#NPP_MIAMI gDM/m2/year To tn DM/ha/year
NPP_MIAMI_tnDM_Ha_Year<-NPP_MIAMI*(1/100)

#NPP_MIAMI tn DM/ha/year To tn C/ha/year
NPP_MIAMI_tnC_Ha_Year<-NPP_MIAMI_tnDM_Ha_Year*0.5

# Save NPP MIAMI MODEL tnC/ha/year
setwd(WD_NPP)
writeRaster(NPP_MIAMI_tnC_Ha_Year,
  filename="NPP_MIAMI_tnC_Ha_Year_STACK_81-00_MAX.tif",
  format="GTiff",overwrite=TRUE)

# NPP MEAN
NPP_MIAMI_MEAN_81_00<-mean(NPP_MIAMI_tnC_Ha_Year)

# Crop & and mask
setwd(WD_NPP)
```

Monthly Vegetation Cover. Google Earth Engine.

Input	Output
Country polygon geometry	NDVI_2015-2019_prop_gt_06_ [country_code]_MONTH_ [Number of the month] (12 Layers to be saved in a googledrive account)

```
NPP_MIAMI_MEAN_81_00_AOI<-crop(NPP_MIAMI_MEAN_81_00,AOI)
NPP_MIAMI_MEAN_81_00_AOI<-resample(NPP_MIAMI_MEAN_81_00_AOI,
                                   SOC_MAP_AOI)
NPP_MIAMI_MEAN_81_00_AOI<-mask(NPP_MIAMI_MEAN_81_00_AOI,AOI)

writeRaster(NPP_MIAMI_MEAN_81_00_AOI,
  filename="NPP_MIAMI_MEAN_81-00_AOI_MAX.tif",
  format="GTiff",overwrite=TRUE)
writeRaster(NPP_MIAMI_MEAN_81_00,
  filename="NPP_MIAMI_MEAN_81-00_MAX.tif",
  format="GTiff",overwrite=TRUE)
```

9.2.6 Script Number 6. "Monthly_vegetation_cover" from Google Earth Engine.

Script number 6 is a Google Earth Engine script. It is aimed at estimating an average vegetation cover status for each month of the year. It estimates, within a specified time series, the probability for each pixel to present NDVI values greater than a specified threshold, over which the soil is vegetated (for example NDVI > 0.6). The result will vary between 0 and 1. Users may modify the time series and NDVI threshold as desired and according to local knowledge.

In order to run the script, the instructions shown for Script number 1 should be followed to select the Area of Interest (AOI). The script is run one time. The user will need to specify the name of the output folder and the name of the output raster. The following lines need to be edited:

- Line 5, the name of the folder where the output raster file is to be saved (in the Google Drive Account);

- Line 6, the name of the country;
- At the end of the document, in the indicated lines: name folder where the output raster file is to be saved (in the Google Drive Account) and name output file

```
//Google Earth Engine
// Monthly Vegetation Cover for Roth C Model
 // Provide a polygon geometry
 // Select the Modis dataset. MOD13A2 is an NDVI product.
 var folder='inputs_NDVI'//change folder
var country='country'//change
var threshold =4000 //NDVI threshold
var maxpx= 267685348

var ndviVis = {
  min: 0.0,
  max: 1.0,
  palette: [
    'FFFFFF', 'CE7E45', 'DF923D', 'F1B555', 'FCD163',
    '99B718', '74A901', '66A000', '529400', '3E8601',
    '207401', '056201', '004C00', '023B01', '012E01',
    '011D01', '011301'
 ],
};

// January
var dataset = ee.ImageCollection('MODIS/006/MOD13A2')
        .filter(ee.Filter.date('2015-01-01', '2020-12-31'))
        .filter(ee.Filter.calendarRange(1,1,'month'));

var ndvi = dataset.select('NDVI');

var mask06= function(image) {

  var mask = image.select('NDVI').gt(threshold);

  return image.updateMask(mask);
  };
```

```
var ndvi_06=ndvi.map(mask06);

var ndvi_06_nn=ndvi_06.reduce(ee.Reducer.count());

var ndvi_nn=ndvi.reduce(ee.Reducer.count());

var prop_cover01= ndvi_06_nn.divide(ndvi_nn);

var Recorte = prop_cover01.clip(geometry);
Map.addLayer(Recorte, ndviVis, 'Recorte_CR_01')

var regionJSON = JSON.stringify(Recorte.getInfo());
Export.image.toDrive({
          image: Recorte.select("NDVI_count"),
          folder: folder,
          description: country+'_NDVI_2015-2020_prop_gt04_M01',
          scale: 1000,
          region:geometry,
          maxPixels: maxpx

});

//// Febraury
var dataset = ee.ImageCollection('MODIS/006/MOD13A2')
          .filter(ee.Filter.date('2015-01-01', '2020-12-31'))
          .filter(ee.Filter.calendarRange(2,2,'month'));

var ndvi = dataset.select('NDVI');


var mask06= function(image) {

  var mask = image.select('NDVI').gt(threshold);

  return image.updateMask(mask);
  };

var ndvi_06=ndvi.map(mask06);
```

```
var ndvi_06_nn=ndvi_06.reduce(ee.Reducer.count());

var ndvi_nn=ndvi.reduce(ee.Reducer.count());

var prop_cover02= ndvi_06_nn.divide(ndvi_nn);


var Recorte = prop_cover02.clip(geometry);
Map.addLayer(Recorte, ndviVis, 'Recorte_CR_02')

var regionJSON = JSON.stringify(Recorte.getInfo());
Export.image.toDrive({
          image: Recorte.select("NDVI_count"),
          folder: folder,
          description: country+'_NDVI_2015-2020_prop_gt04_M02',
          scale: 1000,
          region:geometry,
          maxPixels: maxpx

});

//// March
var dataset = ee.ImageCollection('MODIS/006/MOD13A2')
          .filter(ee.Filter.date('2015-01-01', '2020-12-31'))
          .filter(ee.Filter.calendarRange(3,3,'month'));

var ndvi = dataset.select('NDVI');

var mask06= function(image) {

  var mask = image.select('NDVI').gt(threshold);

  return image.updateMask(mask);
  };

var ndvi_06=ndvi.map(mask06);

var ndvi_06_nn=ndvi_06.reduce(ee.Reducer.count());
```

```
var ndvi_nn=ndvi.reduce(ee.Reducer.count());

var prop_cover03= ndvi_06_nn.divide(ndvi_nn);

var Recorte = prop_cover03.clip(geometry);
Map.addLayer(Recorte, ndviVis, 'Recorte_CR_03')

var regionJSON = JSON.stringify(Recorte.getInfo());
Export.image.toDrive({
          image: Recorte.select("NDVI_count"),
          folder: folder,
          description: country+'_NDVI_2015-2020_prop_gt04_M03',
          scale: 1000,
          region:geometry,
          maxPixels: maxpx

});

/// April
var dataset = ee.ImageCollection('MODIS/006/MOD13A2')
          .filter(ee.Filter.date('2015-01-01', '2020-12-31'))
          .filter(ee.Filter.calendarRange(4,4,'month'));

var ndvi = dataset.select('NDVI');

var mask06= function(image) {

  var mask = image.select('NDVI').gt(threshold);

  return image.updateMask(mask);
  };

var ndvi_06=ndvi.map(mask06);

var ndvi_06_nn=ndvi_06.reduce(ee.Reducer.count());

var ndvi_nn=ndvi.reduce(ee.Reducer.count());
```

```
var prop_cover04= ndvi_06_nn.divide(ndvi_nn);

var Recorte = prop_cover04.clip(geometry);
Map.addLayer(Recorte, ndviVis, 'Recorte_CR_04')

var regionJSON = JSON.stringify(Recorte.getInfo());
Export.image.toDrive({
          image: Recorte.select("NDVI_count"),
          folder: folder,
          description: country+'_NDVI_2015-2020_prop_gt04_M04',
          scale: 1000,
          region:geometry,
          maxPixels: maxpx

});

/// May
var dataset = ee.ImageCollection('MODIS/006/MOD13A2')
          .filter(ee.Filter.date('2015-01-01', '2020-12-31'))
          .filter(ee.Filter.calendarRange(5,5,'month'));

var ndvi = dataset.select('NDVI');

var mask06= function(image) {

  var mask = image.select('NDVI').gt(threshold);

  return image.updateMask(mask);
  };

var ndvi_06=ndvi.map(mask06);

var ndvi_06_nn=ndvi_06.reduce(ee.Reducer.count());

var ndvi_nn=ndvi.reduce(ee.Reducer.count());

var prop_cover05= ndvi_06_nn.divide(ndvi_nn);
```

```
var Recorte = prop_cover05.clip(geometry);
Map.addLayer(Recorte, ndviVis, 'Recorte_CR_05')

var regionJSON = JSON.stringify(Recorte.getInfo());
Export.image.toDrive({
          image: Recorte.select("NDVI_count"),
          folder: folder,
          description: country+'_NDVI_2015-2020_prop_gt04_M05',
          scale: 1000,
          region:geometry,
          maxPixels: maxpx

});

// June
var dataset = ee.ImageCollection('MODIS/006/MOD13A2')
          .filter(ee.Filter.date('2015-01-01', '2020-12-31'))
          .filter(ee.Filter.calendarRange(6,6,'month'));

var ndvi = dataset.select('NDVI');

var mask06= function(image) {

  var mask = image.select('NDVI').gt(threshold);

  return image.updateMask(mask);
  };

var ndvi_06=ndvi.map(mask06);

var ndvi_06_nn=ndvi_06.reduce(ee.Reducer.count());

var ndvi_nn=ndvi.reduce(ee.Reducer.count());

var prop_cover06= ndvi_06_nn.divide(ndvi_nn);

var Recorte = prop_cover06.clip(geometry);
Map.addLayer(Recorte, ndviVis, 'Recorte_CR_06')
```

```
var regionJSON = JSON.stringify(Recorte.getInfo());
Export.image.toDrive({
          image: Recorte.select("NDVI_count"),
          folder: folder,
          description: country+'_NDVI_2015-2020_prop_gt04_M06',
          scale: 1000,
          region:geometry,
          maxPixels: maxpx

});

// July
var dataset = ee.ImageCollection('MODIS/006/MOD13A2')
          .filter(ee.Filter.date('2015-01-01', '2020-12-31'))
          .filter(ee.Filter.calendarRange(7,7,'month'));

var ndvi = dataset.select('NDVI');

var mask06= function(image) {

  var mask = image.select('NDVI').gt(threshold);

  return image.updateMask(mask);
  };

var ndvi_06=ndvi.map(mask06);

var ndvi_06_nn=ndvi_06.reduce(ee.Reducer.count());

var ndvi_nn=ndvi.reduce(ee.Reducer.count());

var prop_cover07= ndvi_06_nn.divide(ndvi_nn);

var Recorte = prop_cover07.clip(geometry);
Map.addLayer(Recorte, ndviVis, 'Recorte_CR_07')
```

```
var regionJSON = JSON.stringify(Recorte.getInfo());
Export.image.toDrive({
          image: Recorte.select("NDVI_count"),
          folder: folder,
          description: country+'_NDVI_2015-2020_prop_gt04_M07',
          scale: 1000,
          region:geometry,
          maxPixels: maxpx

});

// August
var dataset = ee.ImageCollection('MODIS/006/MOD13A2')
          .filter(ee.Filter.date('2015-01-01', '2020-12-31'))
          .filter(ee.Filter.calendarRange(8,8,'month'));

var ndvi = dataset.select('NDVI');

var mask06= function(image) {

  var mask = image.select('NDVI').gt(threshold);

  return image.updateMask(mask);
  };

var ndvi_06=ndvi.map(mask06);

var ndvi_06_nn=ndvi_06.reduce(ee.Reducer.count());

var ndvi_nn=ndvi.reduce(ee.Reducer.count());

var prop_cover08= ndvi_06_nn.divide(ndvi_nn);

var Recorte = prop_cover08.clip(geometry);
Map.addLayer(Recorte, ndviVis, 'Recorte_CR_08')


var regionJSON = JSON.stringify(Recorte.getInfo());
```

```
Export.image.toDrive({
          image: Recorte.select("NDVI_count"),
          folder: folder,
          description: country+'_NDVI_2015-2020_prop_gt04_M08',
          scale: 1000,
          region:geometry,
          maxPixels: maxpx

});

// September
var dataset = ee.ImageCollection('MODIS/006/MOD13A2')
          .filter(ee.Filter.date('2015-01-01', '2020-12-31'))
          .filter(ee.Filter.calendarRange(9,9,'month'));

var ndvi = dataset.select('NDVI');

var mask06= function(image) {

  var mask = image.select('NDVI').gt(threshold);

  return image.updateMask(mask);
  };

var ndvi_06=ndvi.map(mask06);

var ndvi_06_nn=ndvi_06.reduce(ee.Reducer.count());

var ndvi_nn=ndvi.reduce(ee.Reducer.count());

var prop_cover09= ndvi_06_nn.divide(ndvi_nn);

var Recorte = prop_cover09.clip(geometry);
Map.addLayer(Recorte, ndviVis, 'Recorte_CR_09')

var regionJSON = JSON.stringify(Recorte.getInfo());
Export.image.toDrive({
          image: Recorte.select("NDVI_count"),
```

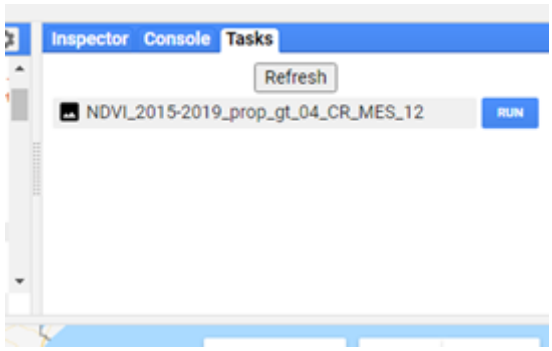
```
          folder: folder,
          description: country+'_NDVI_2015-2020_prop_gt04_M09',
          scale: 1000,
          region:geometry,
          maxPixels: maxpx

});

//  October
var dataset = ee.ImageCollection('MODIS/006/MOD13A2')
          .filter(ee.Filter.date('2015-01-01', '2020-12-31'))
          .filter(ee.Filter.calendarRange(10,10,'month'));

var ndvi = dataset.select('NDVI');

var mask06= function(image) {

  var mask = image.select('NDVI').gt(threshold);

  return image.updateMask(mask);
  };

var ndvi_06=ndvi.map(mask06);

var ndvi_06_nn=ndvi_06.reduce(ee.Reducer.count());

var ndvi_nn=ndvi.reduce(ee.Reducer.count());

var prop_cover10= ndvi_06_nn.divide(ndvi_nn);

var Recorte = prop_cover10.clip(geometry);
Map.addLayer(Recorte, ndviVis, 'Recorte_CR_10')

var regionJSON = JSON.stringify(Recorte.getInfo());
Export.image.toDrive({
          image: Recorte.select("NDVI_count"),
          folder: folder,
          description: country+'_NDVI_2015-2020_prop_gt04_M10',
```

```
          scale: 1000,
          region:geometry,
          maxPixels: maxpx

});

// November
var dataset = ee.ImageCollection('MODIS/006/MOD13A2')
            .filter(ee.Filter.date('2015-01-01', '2020-12-31'))
            .filter(ee.Filter.calendarRange(11,11,'month'));

var ndvi = dataset.select('NDVI');

var mask06= function(image) {

  var mask = image.select('NDVI').gt(threshold);

  return image.updateMask(mask);
  };

var ndvi_06=ndvi.map(mask06);

var ndvi_06_nn=ndvi_06.reduce(ee.Reducer.count());

var ndvi_nn=ndvi.reduce(ee.Reducer.count());

var prop_cover11= ndvi_06_nn.divide(ndvi_nn);

var Recorte = prop_cover11.clip(geometry);
Map.addLayer(Recorte, ndviVis, 'Recorte_CR_11')

var regionJSON = JSON.stringify(Recorte.getInfo());
Export.image.toDrive({
          image: Recorte.select("NDVI_count"),
          folder: folder,
          description: country+'_NDVI_2015-2020_prop_gt04_M11',
          scale: 1000,
          region:geometry,
```

```
          maxPixels: maxpx

});

/// December
var dataset = ee.ImageCollection('MODIS/006/MOD13A2')
            .filter(ee.Filter.date('2015-01-01', '2020-12-31'))
            .filter(ee.Filter.calendarRange(12,12,'month'));

var ndvi = dataset.select('NDVI');

var mask06= function(image) {

  var mask = image.select('NDVI').gt(threshold);

  return image.updateMask(mask);
  };

var ndvi_06=ndvi.map(mask06);

var ndvi_06_nn=ndvi_06.reduce(ee.Reducer.count());

var ndvi_nn=ndvi.reduce(ee.Reducer.count());

var prop_cover12= ndvi_06_nn.divide(ndvi_nn);

var Recorte = prop_cover12.clip(geometry);
Map.addLayer(Recorte, ndviVis, 'Recorte_CR_12')

var regionJSON = JSON.stringify(Recorte.getInfo());
Export.image.toDrive({
          image: Recorte.select("NDVI_count"),
          folder: folder, // change folder name
          description: country+'_NDVI_2015-2020_prop_gt04_M12',
          scale: 1000,
          region:geometry,
          maxPixels: maxpx
```

```
});
```

After running the script, the layer must be saved to the Google Drive account. To accomplish this, the user will need to click on the "task" button and then click the "run" button (Fig. 9.3)



Saving the task in GEE.

Once the procedure is completed, the layers should be downloaded from the Google Drive and saved into a local folder

9.2.7 Script Number 7. "Vegetation_Cover_stack.R"

The script number 7 is an R script that uses the monthly vegetation cover layers (0-1 values) created with the GEE script number 6 to create a raster stack. It also linearly rescales the values from "0 to 1" (proportion of vegetated pixels in a time series) to "1 to 0.6" (being 1 = bare soil and 0.6 = full vegetated pixel). This transformation will allow us to use the calculated values as modifying factors of the decomposition rates in the RothC model.

Once the monthly vegetation cover layers are downloaded from Google Drive, we will generate a stack of those layers. We will first open script number 7 "Vegetation_Cover_stack.R" and the required packages. Then, we will need to open the country polygon vector and set the working directory for the input and the output layers.

Monthly Vegetation Cover Stack.

Input	Output
Country polygon geometry FAO SOC map (master layer): GSOCmapV1.5.0.tif 12x NDVI_2015-2019_prop_gt_06_[country_code]_MONTH_[Number of the month]	Cov_Stack_[country_code].tif (12 layer stack)

```
library(raster)
library(rgdal)
WD_AOI<-("C:/Training_Material/INPUTS/AOI_POLYGON")
WD_SOC<-("C:/Training_Material/INPUTS/SOC_MAP")
WD_COV<-("C:/Training_Material/INPUTS/COV")


# Open the shapefile of the region/country
setwd(WD_AOI)
AOI<-readOGR("Departamento_Pergamino.shp")
#Open SOC MAP FAO
setwd(WD_SOC)
SOC_MAP_AOI<-raster("SOC_MAP_AOI.tif")
# Open Vegetation Cover layer based only
# in proportion of NDVI pixels grater than 0.6
setwd(WD_COV)
Cov1<-raster("Pergamino_NDVI_2015-2020_prop_gt04_M01.tif")
Cov1[is.na(Cov1[])] <- 0
Cov1_crop<-crop(Cov1,AOI)
Cov1_mask<-mask(Cov1_crop,AOI)
Cov1_res<-resample(Cov1_mask,SOC_MAP_AOI,method='ngb')
Cov2<-raster("Pergamino_NDVI_2015-2020_prop_gt04_M02.tif")
Cov2[is.na(Cov2[])] <- 0
Cov2_crop<-crop(Cov2,AOI)
Cov2_mask<-mask(Cov2_crop,AOI)
Cov2_res<-resample(Cov2_mask,SOC_MAP_AOI,method='ngb')
Cov3<-raster("Pergamino_NDVI_2015-2020_prop_gt04_M03.tif")
```

```
Cov3[is.na(Cov3[])] <- 0
Cov3_crop<-crop(Cov3,AOI)
Cov3_mask<-mask(Cov3_crop,AOI)
Cov3_res<-resample(Cov3_mask,SOC_MAP_AOI,method='ngb')
Cov4<-raster("Pergamino_NDVI_2015-2020_prop_gt04_M04.tif")
Cov4[is.na(Cov4[])] <- 0
Cov4_crop<-crop(Cov4,AOI)
Cov4_mask<-mask(Cov4_crop,AOI)
Cov4_res<-resample(Cov4_mask,SOC_MAP_AOI,method='ngb')
Cov5<-raster("Pergamino_NDVI_2015-2020_prop_gt04_M05.tif")
Cov5[is.na(Cov5[])] <- 0
Cov5_crop<-crop(Cov5,AOI)
Cov5_mask<-mask(Cov5_crop,AOI)
Cov5_res<-resample(Cov5_mask,SOC_MAP_AOI,method='ngb')
Cov6<-raster("Pergamino_NDVI_2015-2020_prop_gt04_M06.tif")
Cov6[is.na(Cov6[])] <- 0
Cov6_crop<-crop(Cov6,AOI)
Cov6_mask<-mask(Cov6_crop,AOI)
Cov6_res<-resample(Cov6_mask,SOC_MAP_AOI,method='ngb')
Cov7<-raster("Pergamino_NDVI_2015-2020_prop_gt04_M07.tif")
Cov7[is.na(Cov7[])] <- 0
Cov7_crop<-crop(Cov7,AOI)
Cov7_mask<-mask(Cov7_crop,AOI)
Cov7_res<-resample(Cov7_mask,SOC_MAP_AOI,method='ngb')
Cov8<-raster("Pergamino_NDVI_2015-2020_prop_gt04_M08.tif")
Cov8[is.na(Cov8[])] <- 0
Cov8_crop<-crop(Cov8,AOI)
Cov8_mask<-mask(Cov8_crop,AOI)
Cov8_res<-resample(Cov8_mask,SOC_MAP_AOI,method='ngb')
Cov9<-raster("Pergamino_NDVI_2015-2020_prop_gt04_M09.tif")
Cov9[is.na(Cov9[])] <- 0
Cov9_crop<-crop(Cov9,AOI)
Cov9_mask<-mask(Cov9_crop,AOI)
Cov9_res<-resample(Cov9_mask,SOC_MAP_AOI,method='ngb')
Cov10<-raster("Pergamino_NDVI_2015-2020_prop_gt04_M10.tif")
Cov10[is.na(Cov10[])] <- 0
Cov10_crop<-crop(Cov10,AOI)
Cov10_mask<-mask(Cov10_crop,AOI)
```

```
Cov10_res<-resample(Cov10_mask,SOC_MAP_AOI,method='ngb')
Cov11<-raster("Pergamino_NDVI_2015-2020_prop_gt04_M11.tif")
Cov11[is.na(Cov11[])] <- 0
Cov11_crop<-crop(Cov11,AOI)
Cov11_mask<-mask(Cov11_crop,AOI)
Cov11_res<-resample(Cov11_mask,SOC_MAP_AOI,method='ngb')
Cov12<-raster("Pergamino_NDVI_2015-2020_prop_gt04_M12.tif")
Cov12[is.na(Cov12[])] <- 0
Cov12_crop<-crop(Cov12,AOI)
Cov12_mask<-mask(Cov12_crop,AOI)
Cov12_res<-resample(Cov12_mask,SOC_MAP_AOI,method='ngb')
Stack_Cov<-stack(Cov1_res,Cov2_res,Cov3_res,Cov4_res,
                 Cov5_res,Cov6_res,Cov7_res,Cov8_res,
                 Cov9_res,Cov10_res,Cov11_res,Cov12_res)
# rescale values to 1 if it is bare soil
# and 0.6 if it is vegetated.
Cov<-((Stack_Cov)*(-0.4))+1
writeRaster(Cov,filename='Cov_stack_AOI.tif',format='GTiff')
```

Once the monthly vegetation cover layers are downloaded from Google Drive, we will generate a stack of those layers. We will first open script number 7 "Vegetation_Cover_stack.R" and the required packages. Then, we will need to open the country polygon vector and set the working directory for the input and the output layers.

9.2.8 Script Number 8. "ISRIC_Clay_Layer.R"

ISRIC clay layers represent the clay content (0-2 micrometer; in g/100g; w%) at four standard depths (sl1=0-1cm; sl2=1-5; sl3=5-15cm; sl4=15-30 cm) at a 250m resolution. The objective of this script is to aggregate the different layers into one layer by estimating the weighted average of the four depths:

```
library(raster)
library(rgdal)

WD_AOI<-("C:/Training_Material/INPUTS/AOI_POLYGON")
WD_ISRIC<-("C:/Training_Material/INPUTS/CLAY")
```

Clay layers from ISRIC

Input	Output
Country polygon geometry Clay inputs from ISRIC: CLYPPT_M_sl1_250m_ll.tif CLYPPT_M_sl2_250m_ll.tif CLYPPT_M_sl3_250m_ll.tif CLYPPT_M_sl4_250m_ll.tif	Clay_[country_code]_Avg.tif (1 layer)

```
WD_CLAY<-("C:/Training_Material/INPUTS/CLAY")
# Open the shapefile of the region/country
setwd(WD_AOI)
AOI<-readOGR("Departamento_Pergamino.shp")
# Open Clay layers  (ISRIC)
setwd(WD_ISRIC)
Clay1<-raster("CLYPPT_M_sl1_250m_ll_subs.tif")
Clay2<-raster("CLYPPT_M_sl2_250m_ll_subs.tif")
Clay3<-raster("CLYPPT_M_sl3_250m_ll_subs.tif")
Clay4<-raster("CLYPPT_M_sl4_250m_ll_subs.tif")
Clay1_AR<-crop(Clay1,AOI)
Clay2_AR<-crop(Clay2,AOI)
Clay3_AR<-crop(Clay3,AOI)
Clay4_AR<-crop(Clay4,AOI)
# Average of four depths
WeightedAverage<-function(r1,r2,r3,r4){
  return(r1*(1/30)+r2*(4/30)+r3*(10/30)+r4*(15/30))}
Clay_WA<-overlay(Clay1_AR,Clay2_AR,
                 Clay3_AR,Clay4_AR,fun=WeightedAverage)
Clay_WA_AOI<-mask(Clay_WA,AOI)
setwd(WD_CLAY)
writeRaster(Clay_WA_AOI,filename="Clay_WA_AOI.tif",
            format='GTiff', overwrite = TRUE)
```

9.3 Preparing the land use layer

The land use layer is one of the most important layers in the process, as it defines the target areas and production systems to be modeled. The land use layer will be needed:

- to account for major land use changes during the 2000-2020 period;
- to obtain the DPM/RPM ratios required in the RothC model (See Chapter 4);
- to define the modeling units/target points where the model is to be run (agricultural lands in 2020).

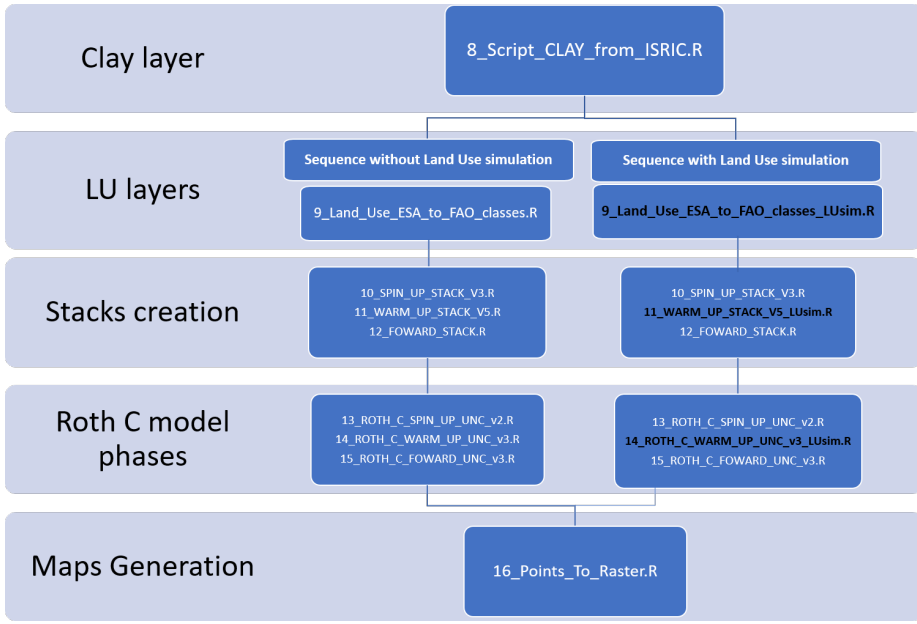
Each modeling phase will require specific land use layers. For the 'spin up' phase, users should use a representative land use layer for the period 1980-2000 (e.g. land use layer as in year 2000), or best available land use layer. For the 'warm-up' phase, users can use year to year land use layers (2000 to 2020), or a representative land use layer for the period, depending on the available information. The 'warm-up' land use layer accounts for year to year changes in the land use during the period (for example a pixel that changes from forest to cropland). The script will need a stack of land use layers, one layer for each year of the warm up phase. If the user does not want to model changes in the land use layer over the warm up phase, or information is not available, the same land use layer for each year can be used over the warm-up phase. For the 'forward' phase, the latest best available land use layer should be used. As a minimum, the last available land use data at 1x1 km resolution shall be defined. The predominant land use category in each cell of the 1x1 km grid shall be selected if finer resolutions are available. The land use classes can be derived from land cover classes from different national, regional or global datasets which best correlate with national land use. The land use layers are used in the three modelling phases to generate a decomposition rate DR layer (generated through scripts 10, 11, and 12, See sections 9.8-9.10), that represents the above mentioned DPM/RPM ratios for the different land use classes. In scripts 10,11 and 12, default DPM/RPM values are assigned to each FAO Global Land Cover (GLC-SHARE) class (See Table 6.1 Chapter 6; Section 6.7; and scripts 10,11 and 12). For more information on this classification refer to FAO (2014) and to the FAO Land and Water site: http://www.fao.org/land-water/land/land-governance/land-resources-planning-toolbox/category/details/en/c/1036355/ Thus, land cover classes obtained from different datasets (e.g. European Space Agency - ESA) need to

be re-classified into FAO land cover classes in a Geotiff format if the scripts 10,11 and 12 are to be run with the default land classes and DPM/RPM ratios provided with the training material. In this section, we provide a script to transform ESA land use cover classes to FAO land use classes (script 9), which can be used as a model to convert and use classes from other datasets. Users can however modify the DPM/RPM default values (See Table 6.1, Chapter 6) based on local knowledge and available information, create additional land use classes or disaggregate the FAO land use classes, and assign DPM/RPM ratios to those new classes by modifying the provided scripts. Users are encouraged to leverage available local knowledge and data to produce the most accurate SOCseq maps possible. With this in mind, if more detailed land use maps, i.e. containing information about the types of cropping systems present, and local data on the DPM/RPM for the specific land use types are easily accessible, the provided script should be edited accordingly.
Finally, the land use layer is also needed to define the target points where the three phases of the protocol will be run. In section 9.7 we provide a Qgis model to generate the target points from the land use layer. Defining the target points out of the land use layer will allow us to run the model just in the pixels with the land use classes of interest.

Depending on whether yearly land use layers are available for the forward phase, this technical manual contains alternative scripts both for the data preparation phase (Scripts 9_Land_Use_ESA_to_FAO_classes_LUsim.R and 11_WARM_UP_STACK_V5_LUsim.R) and the modelling phase (Script 14_ROTH_C_WARM_UP_UNC_v3_LUsim.R). Figure illustrates the script sequence to be followed depending on whether yearly land use change layers are available for the warm up phase.

9.3.1 Script Number 9 "Land_Use_ESA_to_FAO_classes.R" No land use change

Script number 9 transforms the ESA (European Space Agency 2015; 300 m resolution; ESA CCI Land cover website) land cover classes to the FAO land use classes. This script can be modified to be used with any other land use dataset.



Script sequence based on whether yearly land use change layers for the warm up phase are available

Conversion of ESA land use to FAO land use classes

Input	Output
Country polygon geometry FAO SOC map (master layer): GSOCmapV1.5.0.tif ESA Land use: ESACCI-LC-L4-LCCS-Map-300m-P1Y-2015-v2.0.7.tif	ESA_Land_Cover_12clases_FAO_s.tif (1 layer)

First, we will need to open the R packages, open the shapefile of the region/country to be modelled, and open the land use/land cover data set to be re-classified into FAO land use classes:

```
library(raster)
library(rgdal)
WD_AOI<-("C:/Training_Material/INPUTS/AOI_POLYGON")
WD_LU<-("C:/Training_Material/INPUTS/LAND_USE")
WD_SOC<-("C:/Training_Material/INPUTS/SOC_MAP")
# Open the shapefile of the region/country
setwd(WD_AOI)
AOI<-readOGR("Departamento_Pergamino.shp")
# Open Land Use Layer (ESA)
setwd(WD_LU)
ESA_LU<-raster("ESACCI-LC-L4-LCCS-Map-
               300m-P1Y-2015-v2.0.7_subs.tif")
plot(ESA_LU)
# Cut the LU layer by the country polygon
ESA_LU_AOI<-crop(ESA_LU,AOI)
plot(ESA_LU_AOI)
# Reclassify ESA LAND USE to FAO LAND USE classes
#     0 = 0   No Data
#   190 = 1 Artificial
#   10 11 20 30 40 = 2 Croplands
#   130 = 3 Grassland
#   50 60 61 62 70 71 72 80 81 82 90 100 110 = 4 Tree Covered
#   120 121 122= 5 Shrubs Covered
```

```
#   160 180 = 6 Herbaceous vegetation flooded
#   170 = 7 Mangroves
#   150 151 152 153= 8 Sparse Vegetation
#   200 201 202 = 9 Baresoil
#   220 = 10 Snow and Glaciers
#   210 = 11 Waterbodies
#   12 = 12 Treecrops
#   20 = 13 Paddy fields(rice/ flooded crops)
# Reclassify matrix. "Is" to "become"
is<-c(0,190,10,11,20,30,40,130,50,60,61,62,70,71,72,
      80,81,82,90,100,110,120,121,122,160,180,170,
      150,151,152,153,200,201,202,220,210,12)
become<-c(0,1,2,2,2,2,2,3,4,4,4,4,4,4,4,4,4,4,
          4,4,4,5,5,5,6,6,7,8,8,8,8,9,9,9,10,11,12)
recMat<-matrix(c(is,become),ncol=2,nrow=37)
# Reclassify
ESA_FAO <- reclassify(ESA_LU_AOI, recMat)
# Resample to SOC map layer extent and resolution
setwd(WD_SOC)
SOC_MAP_AOI<-raster("SOC_MAP_AOI.tif")
ESA_FAO_res<-resample(ESA_FAO,SOC_MAP_AOI,method='ngb')
ESA_FAO_mask<-mask(ESA_FAO_res,SOC_MAP_AOI)
# Save Land Use raster
setwd(WD_LU)
writeRaster(ESA_FAO_mask,
  filename="ESA_Land_Cover_12clases_FAO_AOI.tif",
  format='GTiff', overwrite = TRUE)
```

9.3.2 Script Number 9 “Land_Use_ESA_to_FAO_classes _LUsim.R” Land use change simulation

Script number 9 transforms the ESA (European Space Agency 2000 to 2018; 300 m resolution; ESA CCI Land cover website) land cover classes to the FAO land use classes. This script allows for the preparation of a stack with yearly land use layers to simulate land use change during the warm up phase.

```
#DATE: 11-02-2021

# MSc Ing Agr Luciano E Di Paolo
# Dr Ing Agr Guillermo E Peralta

#### Prepare Land Use layer

rm(list = ls())

library(raster)
library(rgdal)

WD_AOI<-(
  "C:/TRAINING_MATERIALS/INPUTS/AOI_POLYGON")

WD_LU<-(
  "C:/TRAINING_MATERIALS/INPUTS/LAND_USE")

WD_SOC<-(
  "C:/TRAINING_MATERIALS/INPUTS/SOC_MAP")

# Open the shapefile of the region/country

setwd(WD_AOI)
AOI<-readOGR("Departamento_Pergamino.shp") # change for your
# own Area of interest

# Open Land Use Layer (ESA)

setwd(WD_LU)
ESA_LU<-stack("ESA_Land_Cover_ESA_Stack.tif")
plot(ESA_LU[[1]])

# Cut the LU layer by the country polygon

ESA_LU_AOI<-crop(ESA_LU,AOI)

plot(ESA_LU_AOI[[1:4]])
```

```
# Reclassify ESA LAND USE to FAO LAND USE classes

#   0 = 0     No Data
#   190 = 1 Artificial
#   10 11 30 40 = 2 Croplands
#   130 = 3 Grassland
#   50 60 61 62 70 71 72 80 81 82 90 100 110 = 4 Tree Covered
#   120 121 122= 5 Shrubs Covered
#   160 180 = 6 Herbaceous vegetation flooded
#   170 = 7 Mangroves
#   150 151 152 153= 8 Sparse Vegetation
#   200 201 202 = 9 Bare soil
#   220 = 10 Snow and Glaciers
#   210 = 11 Waterbodies
#   12  = 12 Tree crops
# 20 = 13 Paddy fields(rice/ flooded crops)

# Create a reclassification matrix. "Is" to "become"

is<-c(0,190,10,11,30,40,130,50,60,61,62,70,71,72,
      80,81,82,90,100,110,120,121,122,160,180,
      170,150,151,152,153,200,201,202,220,210,12,20)

become<-c(0,1,2,2,2,2,3,4,4,4,4,4,4,4,4,4,4,4,
          4,4,5,5,5,6,6,7,8,8,8,8,9,9,9,10,11,12,13)

recMat<-matrix(c(is,become),ncol=2,nrow=37)

# Reclassify

ESA_FAO <- reclassify(ESA_LU_AOI, recMat)

# Resample to SOC map layer extent and resolution
setwd(WD_SOC)
SOC_MAP_AOI<-raster("SOC_MAP_AOI.tif") # change for
  # your own SOC MAP

ESA_FAO_res<-resample(ESA_FAO,SOC_MAP_AOI,method='ngb')
```

```
ESA_FAO_mask<-mask(ESA_FAO_res,SOC_MAP_AOI)

# Save Land Use raster
setwd(WD_LU)
writeRaster(ESA_FAO_mask,
  filename="ESA_Land_Cover_12clases_FAO_Stack_AOI.tif",
  format='GTiff',overwrite=TRUE)

# We save separately the land use from 2018
# to perform the target's points creation
writeRaster(ESA_FAO_mask[[18]],
  filename="ESA_Land_Cover_12clases_FAO_2018_AOI.tif",
  format='GTiff',overwrite=TRUE)
```

9.4 Harmonization of soil, climate and vegetation layers.

Once all soil, climate, vegetation and land use layers are created, they need to be harmonized in order to run the model. The harmonization of layers consists of three steps. First, if the model is to be run for an entire country, layers need to be harmonized to the extents of the country boundaries (country polygon layer extents). Second, a resampling process is required in order to match the spatial resolution to the master layer (SOC FAO layer). Finally, a masking process is required to cut the layer with the vector polygon boundaries. After the harmonization of all layers, we will generate a raster stack of all layers needed to run the model. The harmonization/stacking process will be performed three times (scripts 10,11,12), one for each modelling phase.

9.4.1 Script Number 10. "SPIN_UP_STACK.v3.R"

Script number 10 is intended to harmonize all layers needed to complete phase 1 (long spin-up) of the spatial RothC model. The result of this script is a simple raster stack which contains all the data to perform the spin-up phase. To generate the stack we will need the SOC FAO layer (master layer), the clay layer (from script number 8), the three climate stacks (from script number 1),

the land use layer (from script number 9), and the vegetation cover stack (from script number 7).

Spin-up stack

Input	Output
Country polygon geometry FAO SOC map (master layer): GSOCmapV1.5.0.tif Clay inputs (from script number 8): Clay_[country_code]_Avg.tif (1 layer) TerraClimate layers (from Script number 1): Temp_Stack_81-00_TC.tif Prec_Stack_81-00_TC.tif PET_Stack_81-00_TC.tif Land use layer (from script number 9): ESA_Land_Cover_12clases_FAO_s.tif (1 layer) Vegetation cover layer (from script number 7): Cov_Stack_[country_code].tif (12 layer stack)	Stack_Set_SPIN_UP_ [country_code].tif

First, we will open the required R-packages and a shapefile (polygon) which represents the country boundary. In the script below we will be using an example but when running it (AR), the user will have to replace the file according to the target country. The user can also modify the names of the variables inside the script. However, as these variables will only exist inside the script, it is not necessary.

```
# Prepare the layers for the SPIN-UP process of the Roth C Model.
rm(list = ls())
library(raster)
library(rgdal)
WD_AOI<-("C:/Training_Material/INPUTS/AOI_POLYGON")
WD_SOC<-("C:/Training_Material/INPUTS/SOC_MAP")
```

```
WD_CLAY<-("C:/Training_Material/INPUTS/CLAY")
WD_CLIM<-("C:/Training_Material/INPUTS/TERRA_CLIMATE")
WD_LU<-("C:/Training_Material/INPUTS/LAND_USE")
WD_COV<-("C:/Training_Material/INPUTS/COV")
WD_STACK<-("C:/Training_Material/INPUTS/STACK")
# Open the shapefile of the region/country
setwd(WD_AOI)
AOI<-readOGR("Departamento_Pergamino.shp")

The second step is to load the latest version of
FAO Soil Organic Carbon map layer (Master Layer),
created in script number 0.
#Open SOC MAP FAO
setwd(WD_SOC)
SOC_MAP_AOI<-raster("SOC_MAP_AOI.tif")
 Next, we will open the clay content layer (from script number 8):
# Open Clay layer
setwd(WD_CLAY)
Clay_WA_AOI<-raster("Clay_WA_AOI.tif")
Clay_WA_AOI_res<-resample(Clay_WA_AOI,SOC_MAP_AOI,
                          method='bilinear')
#Clay_AR_Avg<-crop(Clay_AR_Avg,AR)
#Clay_AR_Avg<-mask(Clay_AR_Avg,AR)
#Clay_AR_Avg_res<-resample(Clay_AR_Avg,SOC_MAP_AR,
#                          method='bilinear')
```

Next, we will open the climate raster layers (generated in script number 1). These layers come from the TerraClimate database, but the user can choose local layers if desired, as long as they match the arrangement and format needed for running the model.

```
#Open Precipitation layer
setwd(WD_CLIM)
PREC<-stack("Prec_Stack_81-00_TC.tif")
PREC_AOI<-crop(PREC,AOI)
PREC_AOI<-resample(PREC_AOI,SOC_MAP_AOI)
PREC_AOI<-mask(PREC_AOI,AOI)
PREC_AOI<-stack(PREC_AOI)
```

```
#Open Temperatures layer
TEMP<-stack("Temp_Stack_81-00_TC.tif")
TEMP_AOI<-crop(TEMP,AOI)
TEMP_AOI<-resample(TEMP_AOI,SOC_MAP_AOI)
TEMP_AOI<-mask(TEMP_AOI,AOI)
TEMP_AOI<-stack(TEMP_AOI)
#Open Potential Evapotranspiration layer
PET<-stack("PET_Stack_81-00_TC.tif")
PET_AOI<-crop(PET,AOI)
PET_AOI<-resample(PET_AOI,SOC_MAP_AOI)
PET_AOI<-mask(PET_AOI,AOI)
PET_AOI<-stack(PET_AOI)
```

Next, we will open, resample and mask the land use raster layer to be used in the spin up phase (representative 1980-2000 period).
In this example we will use the ESA land used reclassified into FAO land use classes (script 9).

```
# Open Land Use layer reclassify to FAO classes
# 0 No Data
# 1 Artificial
# 2 Croplands
# 3 Grassland
# 4 Tree Covered
# 5 Shrubs Covered
# 6 Herbaceous vegetation flooded
# 7 Mangroves
# 8 Sparse Vegetation
# 9 Baresoil
# 10 Snow and Glaciers
# 11 Waterbodies
# 12 TreeCrops
# 13 Paddy fields
setwd(WD_LU)
LU_AOI<-raster("ESA_Land_Cover_12clases_FAO_2018_AOI.tif")
Then, we will open the vegetation cover layers
(created in script number 7):
# Open Vegetation Cover layer
```

```
setwd(WD_COV)
Cov_AOI<-stack('Cov_stack_AOI.tif')
```

The script then creates a DR layer (DPM/RPM ratio). Here the DR layer is derived from the Land use layer, assigning default DPM/RPM ratios to each FAO land cover class (See Table 9.13). Users can modify these ratios according to local expertise and available local information.

```
# Use Land use layer to convert it to DR layer
#DPM/RPM (decomposable vs resistant plant material)
#(1) Most agricultural crops and improved grassland
# and tree crops 1.44
#(2) Unimproved grassland and shrub 0.67
#(3) Deciduous and tropical woodland 0.25
DR<-(LU_AOI==2 | LU_AOI==12| LU_AOI==13)*1.44+ (LU_AOI==4)*0.25 +
  (LU_AOI==3 | LU_AOI==5 | LU_AOI==6 | LU_AOI==8)*0.67
Finally, we will create a stack with all the raster layers
that have been prepared.
 # STACK all layers
Stack_Set_AOI<-stack(SOC_MAP_AOI,
    Clay_WA_AOI_res,TEMP_AOI,PREC_AOI,PET_AOI,DR,LU_AOI,Cov_AOI)
setwd(WD_STACK)
writeRaster(Stack_Set_AOI,
    filename=("Stack_Set_SPIN_UP_AOI.tif"),
    format="GTiff", overwrite = TRUE)
```

9.4.2 Script Number 11. "WARM_UP_STACK_V5.R" No Land use change

Script number 11 is intended to harmonize all layers required to run the phase 2 (WARM UP) of the spatial RothC model. The result of this script is a simple raster stack which contains most of the layers needed for the warm-up phase. To generate the stack we will need the latest version of SOC FAO layer (master layer), the clay layer (from script number 8), land use layers (from script number 9), a land use stack (one land use layer per year), a vegetation cover stack (from script number 7) and the NPP stack (from script number 4). The climate layers and the NPP mean are additional layers that will be needed in the WARM UP

phase but will not be part of this stack because of the final size of the output file.

Warm-up stack

Input	Output
Country polygon geometry SOC layer (from script no. 10): SOC_MAP_[country_code].tif Clay inputs (from script no. 8): Clay_[country_code]_Avg.tif (1 layer) Land use layer (from script no. 9): ESA_Land_Cover_12clases_FAO_s.tif (18 layer, 1 per year) Vegetation cover layer (from script no. 7): Cov_Stack_[country_code].tif (12 layer stack)	Stack_Set_WARM_UP_ [country_code].tif

First, we will load the packages, set the number of years of the warm up phase and set the directories of each layer. Then we will open the country vector polygon boundaries:

```
rm(list = ls())
library(raster)
library(rgdal)
# Set the number of years of the warm up
nWUP<-18
WD_AOI<-("C:/Training_Material/INPUTS/AOI_POLYGON")

WD_SOC<-("C:/Training_Material/INPUTS/SOC_MAP")
WD_CLAY<-("C:/Training_Material/INPUTS/INPUTS/CLAY")
WD_CLIM<-("C:/Training_Material/INPUTS/TERRA_CLIMATE")
WD_LU<-("C:/Training_Material/INPUTS/LAND_USE")
WD_COV<-("C:/Training_Material/INPUTS/COV")
WD_STACK<-("C:/Training_Material/INPUTS/STACK")
```

```
WD_NPP<-("C:/Training_Material/INPUTS/NPP")
# Open the shapefile of the region/country
setwd(WD_AOI)
AOI<-readOGR("Departamento_Pergamino.shp")
```

Then, we will open the harmonized FAO GSOCmap of the country created in script number 0:

```
#Open SOC MAP
setwd(WD_SOC)
SOC_MAP_AOI<-raster("SOC_MAP_AOI.tif")
Then we will open the clay layer created in script number 8:
 # Open Clay layers  (ISRIC)
setwd(WD_CLAY)
Clay_WA_AOI<-raster("Clay_WA_AOI.tif")
Clay_WA_AOI_res<-resample(Clay_WA_AOI,
                          SOC_MAP_AOI,method='bilinear')
```

Then, we will open the Land Use layers required for the warm up phase (2000-2020). In this example we used the ESA land use (2015) reclassified into to the FAO land use classes.

```
# OPen Land Use layer (ESA)
setwd(WD_LU)
LU_AOI<-raster("ESA_Land_Cover_12clases_FAO_2018_AOI.tif")
```

We will then open the vegetation cover layer previously created in the script number 7:

```
# Open Vegetation Cover layer
setwd(WD_COV)
Cov_AOI<-stack('Cov_stack_AOI.tif')
```

If year to year land use layers are available for the warm up phase (2000-2020), we will open the Land Use stack of the annual land use layers. If annual land use layers are not available, we will just replicate a representative land use layer for the warm-up phase, as previously loaded.

```
# Open Land Use Stack , One Land use layer for each year (in
# this example we use the same LU for the 18/20 year
# period set previously in the nWUP variable
# here 18 years are replicated for the Warm-up
nWUP <- 18
LU_Stack <-stack(replicate(nWUP, LU_AOI))
```

Then, we will create a "DR" stack layer, one DR layer per year of the WARM UP phase.

```
# Create DR Layer from LU layer (ESA land use , 14 classes)
#DPM/RPM (decomposable vs resistant plant material)
#(1) Most agricultural crops and improved
# grassland or tree crops 1.44
#(2) Unimproved grassland and shrub 0.67
#(3) Deciduous and tropical woodland 0.25
DR<-(LU_AOI==2 | LU_AOI==12| LU_AOI==13)*1.44+ (LU_AOI==4)*0.25 +
  (LU_AOI==3 | LU_AOI==5 | LU_AOI==6 | LU_AOI==8)*0.67
DR_Stack<-LU_Stack
for (i in 1:nlayers(LU_Stack)){
DR_Stack[[i]]<-(LU_Stack[[i]]==2 | LU_Stack[[i]]==12)*1.44 +
  (LU_Stack[[i]]==4)*0.25 + (LU_Stack[[i]]==3 | LU_Stack[[i]]==5
          | LU_Stack[[i]]==6 | LU_Stack[[i]]==8)*0.67
}
```

Finally, we will run the rest of the code and save the raster stack containing all the necessary layers to run the 'warm up' phase.

```
# STACK all layers
Stack_Set_AOI<-stack(SOC_MAP_AOI,Clay_WA_AOI_res,
                     Cov_AOI,LU_Stack,DR_Stack)
setwd(WD_STACK)
writeRaster(Stack_Set_AOI,
            filename=("Stack_Set_WARM_UP_AOI.tif"),
            format="GTiff", overwrite = TRUE)
```

9.4.3 Script Number 11. WARM_UP_STACK_V5_LUsim.R Land use change simulation

The following script is to be used if yearly land use layers to simulate land use change are available. This script uses the output from script numer 9 Land_Use_ESA_to_FAO_classes_LUsim.R

```
#DATE 11-2-2021
# ADD NPP_MIN AND NPP_MAX TO THE STACK TO CALCULATE UNCERTAINTIES

# MSc Ing Agr Luciano E Di Paolo
# Dr Ing Agr Guillermo E Peralta


#### Prepare the layers for the WARM UP Roth C Model.

rm(list = ls())

library(raster)
library(rgdal)

# Set the number of years of the warm up
nWUP<-18

WD_AOI<-(
  "C:/TRAINING_MATERIALS/INPUTS/AOI_POLYGON")

WD_SOC<-("C:/TRAINING_MATERIALS/INPUTS/SOC_MAP")

WD_CLAY<-("C:/TRAINING_MATERIALS/INPUTS/CLAY")

WD_CLIM<-(
  "C:/TRAINING_MATERIALS/INPUTS/TERRA_CLIMATE")

WD_LU<-("C:/TRAINING_MATERIALS/INPUTS/LAND_USE")

WD_COV<-("C:/TRAINING_MATERIALS/INPUTS/COV")
```

```
WD_STACK<-("C:/TRAINING_MATERIALS/INPUTS/STACK")

WD_NPP<-("C:/TRAINING_MATERIALS/INPUTS/NPP")


# Open the shapefile of the region/country
setwd(WD_AOI)
AOI<-readOGR("Departamento_Pergamino.shp") # change the AOI

#Open SOC MAP

setwd(WD_SOC)
SOC_MAP_AOI<-raster("SOC_MAP_AOI.tif") # change the SOC_MAP

# Open Clay layers  (ISRIC)

setwd(WD_CLAY)

Clay_WA_AOI<-raster("Clay_WA_AOI.tif")

Clay_WA_AOI_res<-resample(Clay_WA_AOI,SOC_MAP_AOI,
                          method='bilinear')

# OPen Land Use layer (ESA)

# 0 No Data
# 1 Artificial
# 2 Croplands
# 3 Grassland
# 4 Tree Covered
# 5 Shrubs Covered
# 6 Herbaceous vegetation flooded
# 7 Mangroves
# 8 Sparse Vegetation
# 9 Baresoil
# 10 Snow and Glaciers
# 11 Waterbodies
# 12 TreeCrops
```

```
# 13 Paddy fields


setwd(WD_LU)
LU_AOI<-stack("ESA_Land_Cover_12clases_FAO_Stack_AOI.tif")

# Open Vegetation Cover layer

setwd(WD_COV)
Cov_AOI<-stack('Cov_stack_AOI.tif')

# Open Land Use Stack , One Land use layer for each year
#(in this example we use the same LU for the 18 year period)

#LU_Stack <-stack(replicate(nWUP, LU_AOI))
#LU_Stack <-stack(ESA[2001:2015],2015,2015,2015)
LU_Stack<-LU_AOI

# Convert LU layer  to DR layer (ESA land use , 14 classes)

#DPM/RPM (decomposable vs resistant plant material)
#(1) Most agricultural crops and
# improved grassland or tree crops 1.44
#(2) Unimproved grassland and shrub 0.67
#(3) Deciduous and tropical woodland 0.25

#DR<-(LU_AOI==2 | LU_AOI==12 | LU_AOI==13)*1.44+
#(LU_AOI==4)*0.25 + (LU_AOI==3 | LU_AOI==5 |
# LU_AOI==6 | LU_AOI==8)*0.67

DR_Stack<-LU_Stack

for (i in 1:nlayers(LU_Stack)){
DR_Stack[[i]]<-(LU_Stack[[i]]==2 | LU_Stack[[i]]==12 |
                  LU_Stack[[i]]==13)*1.44+
  (LU_Stack[[i]]==4)*0.25 + (LU_Stack[[i]]==3 |
                LU_Stack[[i]]==5 | LU_Stack[[i]]==6 |
                             LU_Stack[[i]]==8)*0.67
```

```
}

# STACK all layers

Stack_Set_AOI<-stack(SOC_MAP_AOI,
            Clay_WA_AOI_res,Cov_AOI,LU_Stack,DR_Stack)

setwd(WD_STACK)
writeRaster(Stack_Set_AOI,
    filename=("Stack_Set_WARM_UP_AOI.tif"),
    format="GTiff",overwrite=TRUE)
```

9.4.4 Script Number 12. "FORWARD_STACK.R"

Script number 12 harmonizes all layers needed to run phase 3 (forward) of the spatial Roth C model. The result of the script is a simple raster stack which contains the layers needed to perform the forward phase. To generate the stack we will need the SOC FAO layer (master layer), the clay layer (from script number 8), the three climate stacks required for the forward phase (from script number 2), the land use layer or the forward phase (from script number 9), and the vegetation cover stack (from script number 7).

Warm-up stack

Input	Output
Country polygon geometry SOC layer (from script no. 10): SOC_MAP_[country_code].tif Clay layer (from script no. 8): Clay_[country_code]_Avg.tif TerraClimate layers (from Script no. 2): Temp_Stack_81-00_TC.tif Prec_Stack_81-00_TC.tif PET_Stack_81-00_TC.tif Land use layer (from script no. 10): LU_res.tif Vegetation cover layer (from script no. 7): Cov_Stack_[country_code].tif (12 layer stack)	Stack_Set_FORWARD _[country_code].tif

First, we will load the packages, set path to the files directories and open the country vector polygon boundaries.

9.4.4.1 Prepare the layers for the FORWARD Mode Roth C Model.

```
rm(list = ls())
library(raster)
library(rgdal)

WD_AOI<-("C:/Training_Material/INPUTS/AOI_POLYGON")
WD_SOC<-("C:/Training_Material/INPUTS/SOC_MAP")
WD_CLAY<-("C:/Training_Material/INPUTS/CLAY")
WD_CLIM<-("C:/Training_Material/INPUTS/TERRA_CLIMATE")
WD_LU<-("C:/Training_Material/INPUTS/LAND_USE")
WD_COV<-("C:/Training_Material/INPUTS/COV")
```

```
WD_STACK<-("C:/Training_Material/INPUTS/STACK")
# Open the shapefile of the region/country
setwd(WD_AOI)
AOI<-readOGR("Departamento_Pergamino.shp")
Then, we will open the SOC layer and the clay layer.
#Open SOC MAP
setwd(WD_SOC)
SOC_MAP_AOI<-raster("SOC_MAP_AOI.tif")
# Open Clay layers  (ISRIC)
setwd(WD_CLAY)
Clay_WA_AOI<-raster("Clay_WA_AOI.tif")
Clay_WA_AOI_res<-resample(Clay_WA_AOI,SOC_MAP_AOI,
                          method='bilinear')

Then we will open the 2000-2020 average climate layers created
(as the one created in script number 2)
#Open Precipitation layer
setwd(WD_CLIM)
PREC<-stack("Prec_Stack_01-18_TC.tif")
PREC_AOI<-crop(PREC,AOI)
PREC_AOI<-resample(PREC_AOI,SOC_MAP_AOI)
PREC_AOI<-mask(PREC_AOI,AOI)
PREC_AOI<-stack(PREC_AOI)
#Open Temperatures layer
TEMP<-stack("Temp_Stack_01-18_TC.tif")
TEMP_AOI<-crop(TEMP,AOI)
TEMP_AOI<-resample(TEMP_AOI,SOC_MAP_AOI)
TEMP_AOI<-mask(TEMP_AOI,AOI)
TEMP_AOI<-stack(TEMP_AOI)
#Open Potential Evapotranspiration layer
PET<-stack("PET_Stack_01-18_TC.tif")
PET_AOI<-crop(PET,AOI)
PET_AOI<-resample(PET_AOI,SOC_MAP_AOI)
PET_AOI<-mask(PET_AOI,AOI)
PET_AOI<-stack(PET_AOI)
Then, we will open the land use layer (latest available year)
created in script number 10.
setwd(WD_LU)
```

```
LU_AOI<-raster("ESA_Land_Cover_12clases_FAO_2018_AOI.tif")
Then, we will open the vegetation cover
layer created in script number 7.
# Open Vegetation Cover
setwd(WD_COV)
Cov_AOI<-stack('Cov_stack_AOI.tif')
As in the previous scripts, this script creates a
DR layer (DPM/RPM ratio), assigning default DPM/RPM ratios
to each FAO land cover class (See Table 9.13).
Users can modify these ratios according to local expertise
and available local information.
# Open Land use layer and convert it to DR layer
# (mod 12 , 14 classes)
#DPM/RPM (decomposable vs resistant plant material)
#(1) Most agricultural crops and improved
# grassland or tree crops 1.44
#(2) Unimproved grassland and shrub 0.67
#(3) Deciduous and tropical woodland 0.25
DR<-(LU_AOI==2 | LU_AOI==12| LU_AOI==13)*1.44+ (LU_AOI==4)*0.25 +
  (LU_AOI==3 | LU_AOI==5 | LU_AOI==6 | LU_AOI==8)*0.67
```

We will create a stack for the forward modelling phase. We will have to define the filename and save the output stack.

```
# STACK all layers
Stack_Set_AR<-stack(SOC_MAP_AOI,Clay_WA_AOI_res,TEMP_AOI,
                    PREC_AOI,PET_AOI,DR,LU_AOI,Cov_AOI)
setwd(WD_STACK)
writeRaster(Stack_Set_AR,filename=("Stack_Set_FORWARD.tif"),
            format="GTiff")
```

9.5 Defining target points to run the model

At this point we have three raster stacks for the different modelling phases. We need to create the points where those simulations will be run in order to accelerate the modelling process. These points will be the center of the pixels

QGIS Procedure No. 1

Input	Output
Land use layer : (from script number 10) LU_res.tif	Points vector for agriculture, forests, tree crops, grasslands, and shrubland classes (empty)

of the master layer (GSOCmap layer, script number 7). Later, we will convert the points containing the modelling output values back to a raster layer format.

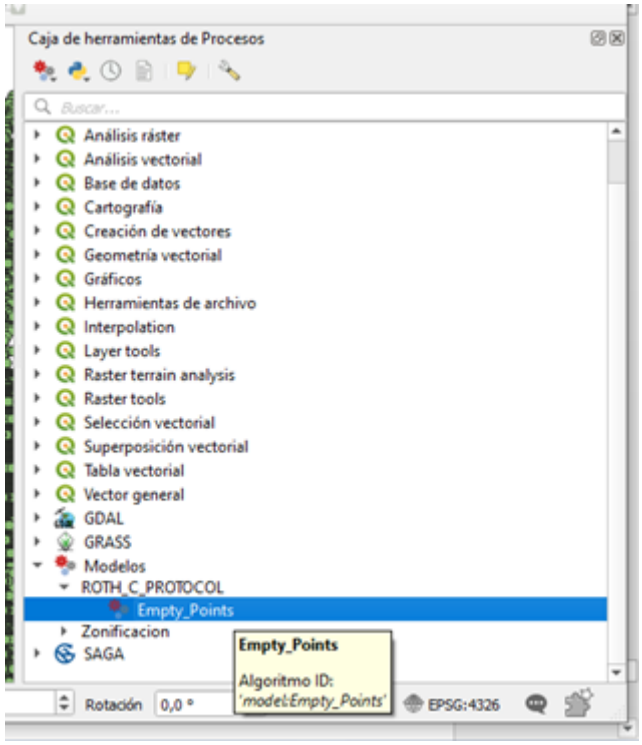
9.5.1 QGIS Procedure number 1 (model)

We will need the land use data of each pixel (we already corregistered the land use layer with the master layer at script number 7). Then we will use the land use layer of the country to generate the points. For this, we can use a QGIS model to create target points.

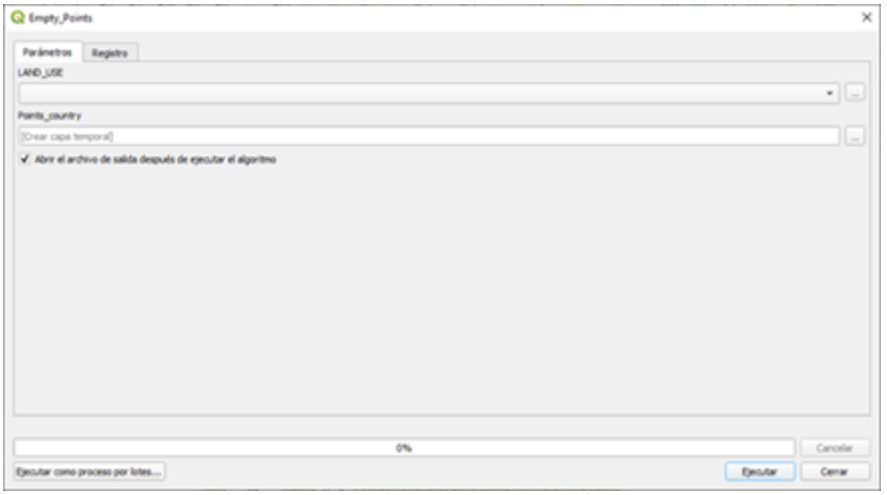
We will open Qgis, go to 'Processing' and select 'Toolbox', and click on the "open existing model" button. We will have to search for the model in the provided folder, called "7_TARGET_POINTS_COUNTRY". We will have to load the model called "Qgis_Procedure_number_1.model3". Once this is done, we can run the model from the processing toolbox.

We will click the Empty_Points button and a window will pop up. We will select the Land use layer created in script number 10 (already resampled to match the extent and pixel size of the GSOCmap), set the path and the name of the output file, and click on the Execute button.

This process will create vector points. Each point will be created in the centroid of each pixel of the land use layer. This vector will contain no fields. The scripts to run the model for each phase (SPIN_UP, WARM_UP, FORWARD) will attach all the necessary data from the stacks (scripts number 10, 11 and 12) to each point.



Processing toolbox in Qgis



Qgis Window to edit the points generated.

Chapter 10

Stage 2: running the model

Once all input layers are prepared, harmonized and stacked, we will run the three modeling phases (spin up, warm up and forward phase). At this stage, we will run the model three times, once for each phase using three different scripts (scripts 13 A (equilibrium run) or 13 B (analytical solution), 14 and 15), which use the same RothC function. For each script we will need the previously created raster stacks and target points. Each script will generate output vector points (containing the modeling results, i.e. SOC stocks of the different carbon pools of the RothC model). The output vector of each phase will be used as an input of the next modeling phase. Finally, after running the final modeling phase, the forward phase, we will obtain an output vector containing the SOC data for each projected scenario. This output vector will be used as input for the final script (script 16) to generate the raster files to build the sequestration potential maps.

10.1 The main commands to perform the RothC calculations

The RothC function is the core of the next three scripts and will be used to simulate the different C pools over the defined time periods. In the following a brief overview of the commands we will be running to perform the RothC calculations will be provided.

```
# ROTH C MODEL FUNCTION .
########## function set up starts###############
Roth_C<-function(Cinputs,years,DPMptf, RPMptf, BIOptf,
      HUMptf, FallIOM,Temp,Precip,Evp,Cov,Cov1,Cov2,
      soil.thick,SOC,clay,DR,bare1){
```

This function will calculate first the "Temperature factor per month" using a function from the SOILR package called "fT.RothC" :

```
#Temperature factor per month
fT=fT.RothC(Temp[,2])
```

Then the function will calculate the "Moisture factor per month" (this function was modified from the original SOILR moisture function, to include the soil cover effect, as in the original RothC model (See Chapter 4):

```
#Moisture effects per month .
fw1func<-function(P, E, S.Thick = 30, pClay = 32.0213,
                  pE = 1, bare)
{

    M = P - E * pE
    Acc.TSMD = NULL
    for (i in 2:length(M)) {
    B = ifelse(bare[i] == FALSE, 1, 1.8)
     Max.TSMD = -(20 + 1.3 * pClay - 0.01 * (pClay^2)) *
       (S.Thick/23) * (1/B)
        Acc.TSMD[1] = ifelse(M[1] > 0, 0, M[1])
        if (Acc.TSMD[i - 1] + M[i] < 0) {
            Acc.TSMD[i] = Acc.TSMD[i - 1] + M[i]
        }
        else (Acc.TSMD[i] = 0)
        if (Acc.TSMD[i] <= Max.TSMD) {
            Acc.TSMD[i] = Max.TSMD
        }
    }
    b = ifelse(Acc.TSMD > 0.444 * Max.TSMD, 1, (0.2 + 0.8 *
          ((Max.TSMD - Acc.TSMD)/(Max.TSMD -
```

```
                                        0.444 * Max.TSMD))))
    b<-clamp(b,lower=0.2)
    return(data.frame(Acc.TSMD, b, Max.TSMD))
}

fW_2<- fw1func(P=(Precip[,2]), E=(Evp[,2]), S.Thick = soil.thick,
               pClay = clay, pE = 1, bare=bare1)$b
```

Then the function will calculate the "vegetation cover factor" effect:

```
#Vegetation Cover effects
fC<-Cov2[,2]
```

In each script, we will need to set the factor frame to run the model (500 years for spin up, 18/20 years for the warm-up, 20 years for the forward), and run the model. We will have two options: one from the SoilR package (using "lsoda" function to solve the differential equations) and one from the soil assessment package that allows to change the differential equation solver, by default we will use the "euler" method, which is faster.

```
# Set the factors frame for Model calculations
xi.frame=data.frame(years,rep(fT*fW_2*fC*fPR,
                    length.out=length(years)))


# RUN THE MODEL from soilassessment
#Roth C soilassesment
Model3_spin=carbonTurnover(tt=years,
  C0=c(DPMptf, RPMptf, BIOptf, HUMptf, FallIOM),
  In=Cinputs,Dr=DR,clay=clay,effcts=xi.frame, "euler")
Ct3_spin=Model3_spin[,2:6]

# RUN THE MODEL FROM SOILR
#Model3_spin=RothCModel(t=years,
# C0=c(DPMptf, RPMptf, BIOptf, HUMptf, FallIOM),
# In=Cinputs,DR=DR,clay=clay,xi=xi.frame, pass=TRUE)
#Ct3_spin=getC(Model3_spin)
```

```
# Get the final pools of the time series
poolSize3_spin=as.numeric(tail(Ct3_spin,1))

return(poolSize3_spin)
}
########## function set up ends###############
```

10.2 Initialization - Spin up phase

To estimate initial carbon pools and equilibrium carbon inputs, two alternatives are provided. Users can run the initialization phase using the equilibrium procedure (explained in section 5.4.1.1) implemented in Script 13 A; or use the analytical procedure (explained in section 5.4.1.2) implemented in Script 13 B. Users may be more familiar with the equilibrium procedure (e.g. Smith et al. 2005; 2006; 2007; Gottschalk et al., 2012) and run Script 13A However, depending on the size of the target area and selected equilibrium period this approach might require a considerable execution time. A minimum of 500 years is suggested to reach equilibrium with reduced computational time. However, it must be noted that spin up runs for 500 years may not necessarily end up in equilibrium SOC stocks, depending on soil, climate and land use conditions. Increasing the duration (1 000-2 000 years) will reduce deviations with the cost of additional computation time. The analytical approach implemented in Script 13.B (see section 10.1.2) was developed as a time-effective and precise alternative to overcome these issues.

10.2.1 Script number 13.A. "ROTH_C_SPIN_UP_UNC_ v2.R" (equilibrium runs)

Script number 13.A implements the first modeling phase (spin up) using the original equilibrium run approach (see section 5.4.1.1). In this script we will load the stack generated in script number 10 and the target points (Section 9.7, QGIS model number 1). We will obtain an output vector containing our target points. This script runs the RothC model for a minimum of 500 years to calculate the equilibrium carbon inputs (the carbon inputs needed to reach the 2001 SOC stocks) and the SOC stocks for the different pools. It first runs using

a standard C input of 1 tC ha^{-1} yr^{-1}, and then equilibrium inputs are estimated from the obtained results and GSOCmap stocks (See Chapters 5 and 6). In this script we will use pedotransfer functions to estimate the SOC stocks of the different pools from the total SOC stock (Weihermüller et al., 2013) to accelerate the spin up process. All that information will be saved to the output vector (shapefile file). The SPIN UP Phase will allow us to calculate two outputs that will be saved to a point vector layer called "C_INPUT_EQ.shp": equilibrium carbon inputs (Ceq) and the carbon stocks of the different soil C pools to run the second phase (WARM UP phase). First, the following packages are loaded into R:

```
rm(list=ls())
library(SoilR)
library(raster)
library(rgdal)
library(soilassessment)
```

Then we will set the working directory.

```
WD_FOLDER<-("C:/TRAINING_MATERIALS")
```

Then, we need to load the target points created in the "qgis procedure number 1".

```
# Vector must be an empty points vector.
setwd(WD_FOLDER)
Vector<-readOGR(
  "INPUTS/TARGET_POINTS/target_points_sub.shp")
```

Then we need to open the stack with all the spin-up variables to run the model (from script number 7).

```
# Stack_Set_1 is a stack that contains the spatial variables
Stack_Set_1<- stack("INPUTS/STACK/Stack_Set_SPIN_UP_AOI.tif")
```

We will run the next lines of the code and create an empty vector variable to save the outputs results of the model and the pedotransfer functions:

```
# Create A vector to save the results
C_INPUT_EQ<-Vector
```

Now, we need to extract the input variables from the raster stack to the target points (now called "Vector"). This step will allow us to continue working with a "dataframe" instead of a raster stack layer.

```
# extract variables to points
Vector_variables<-extract(Stack_Set_1,Vector,df=TRUE)
```

The next lines will define the different variables from the Vector_variables pool. We need to individualize them in separate variables. The last line of this block will set the number of years to run the RothC function.

```
# Extract the layers from the Vector
SOC_im<-Vector_variables[[2]] # first band of the stack
# is the second column of the vector
clay_im<-Vector_variables[[3]]
DR_im<-Vector_variables[[40]]
LU_im<-Vector_variables[[41]]
# Define Years for Cinputs calculations
years=seq(1/12,500,by=1/12)
```

Once we have defined the variables and time frame, we can start using the the RothC function.

```
# ROTH C MODEL FUNCTION .  function set
# up starts###############
Roth_C <- function(Cinputs, years, DPMptf,
    RPMptf, BIOptf, HUMptf, FallIOM, Temp,
    Precip, Evp, Cov, Cov1, Cov2, soil.thick,
    SOC, clay, DR, bare1, LU)  SOC,
    SOC, clay, DR, bare1, LU)  clay,
    SOC, clay, DR, bare1, LU)  DR,
    SOC, clay, DR, bare1, LU)  bare1,
    SOC, clay, DR, bare1, LU)  LU)
{
```

```
# Paddy fields coefficient fPR =
# 0.4 if the target point is class
# = 13 , else fPR=1 From Shirato
# and Yukozawa 2004

fPR = (LU == 13) * 0.4 + (LU != 13) *
    1

# Temperature effects per month
fT = fT.RothC(Temp[, 2])

# Moisture effects per month .

fw1func <- function(P, E, S.Thick = 30,
    pClay = 32.0213, pE = 1, bare) {

    M = P - E * pE
    Acc.TSMD = NULL
    for (i in 2:length(M)) {
        B = ifelse(bare[i] == FALSE,
            1, 1.8)
        Max.TSMD = -(20 + 1.3 * pClay -
            0.01 * (pClay^2)) * (S.Thick/23) *
            (1/B)
        Acc.TSMD[1] = ifelse(M[1] > 0,
            0, M[1])
        if (Acc.TSMD[i - 1] + M[i] <
            0) {
            Acc.TSMD[i] = Acc.TSMD[i -
              1] + M[i]
        } else (Acc.TSMD[i] = 0)
        if (Acc.TSMD[i] <= Max.TSMD) {
            Acc.TSMD[i] = Max.TSMD
        }
    }
    b = ifelse(Acc.TSMD > 0.444 * Max.TSMD,
        1, (0.2 + 0.8 * ((Max.TSMD -
            Acc.TSMD)/(Max.TSMD - 0.444 *
```

```
            Max.TSMD))))
    b <- clamp(b, lower = 0.2)
    return(data.frame(b))
}

fW_2 <- fw1func(P = (Precip[, 2]), E = (Evp[,
    2]), S.Thick = soil.thick, pClay = clay,
    pE = 1, bare = bare1)$b

# Vegetation Cover effects

fC <- Cov2[, 2]

# Set the factors frame for Model
# calculations

xi.frame = data.frame(years, rep(fT *
    fW_2 * fC * fPR, length.out = length(years)))

# RUN THE MODEL from soilassessment
# Roth C soilassesment
Model3_spin = carbonTurnover(tt = years,
    C0 = c(DPMptf, RPMptf, BIOptf, HUMptf,
        FallIOM), In = Cinputs, Dr = DR,
    clay = clay, effcts = xi.frame, "euler")
Ct3_spin = Model3_spin[, 2:6]

# RUN THE MODEL FROM SOILR
# Model3_spin=RothCModel(t=years,C0=c(DPMptf,
# RPMptf, BIOptf, HUMptf,
# FallIOM),In=Cinputs,DR=DR,clay=clay,xi=xi.frame,
# pass=TRUE)
# Ct3_spin=getC(Model3_spin)

# Get the final pools of the time
# series
poolSize3_spin = as.numeric(tail(Ct3_spin,
    1))
```

```
    return(poolSize3_spin)
}
########## function set up
########## ends################
```

After setting the RothC function we will iterate it over each one of the target points.
For each target point we extract the climate variables and the monthly vegetation cover values. Each variable will be assigned to an R variable.

```
# Iterates over the area of interest
########for loop starts################
for (i in 1:dim(Vector_variables)[1]) {
# Extract the variables
Vect<-as.data.frame(Vector_variables[i,])
Temp<-as.data.frame(t(Vect[4:15]))
Temp<-data.frame(Month=1:12, Temp=Temp[,1])
Precip<-as.data.frame(t(Vect[16:27]))
Precip<-data.frame(Month=1:12, Precip=Precip[,1])
Evp<-as.data.frame(t(Vect[28:39]))
Evp<-data.frame(Month=1:12, Evp=Evp[,1])
Cov<-as.data.frame(t(Vect[42:53]))
Cov1<-data.frame(Cov=Cov[,1])
Cov2<-data.frame(Month=1:12, Cov=Cov[,1])
```

The next line will avoid running the model over points with unreliable data, that may contain missing values or unrealistic values.

```
#Avoid calculus over Na values
if (any(is.na(Evp[,2])) | any(is.na(Temp[,2])) |
    any(is.na(SOC_im[i])) | any(is.na(clay_im[i])) |
    any(is.na(Precip[,2]))  |  any(is.na(Cov2[,2]))  |
    any(is.na(Cov1[,1]))  | any(is.na(DR_im[i])) |
    (SOC_im[i]<0) | (clay_im[i]<0)) {C_INPUT_EQ[i,2]<-0}else{
```

Now we will set the value of each variable (SOC, Clay, DR, and Land Use) needed to run the model. The bare1 variable is used to derive the moisture factor.

```
# Set the variables from the images
soil.thick = 30  #Soil thickness (org. layer topsoil), in cm
SOC <- SOC_im[i]  #Soil organic carbon in Mg/ha
clay <- clay_im[i]  #Percent clay %
DR <- DR_im[i]  # DPM/RPM (decomposable vs resistant
# plant material.)
bare1 <- (Cov1 > 0.8)  # If the surface is bare or vegetated
LU <- LU_im[i]
```

The next line will calculate the IOM fraction of the SOC, from the SOC value:

```
#IOM using Falloon method
FallIOM=0.049*SOC^(1.139)
```

Now there are two options to calculate the uncertainties. One is to use your own SOC uncertainty layer: by loading it into R with the following lines of code:

```
# If you use a SOC uncertainty layer
# turn on this. First open the layer
# SOC_UNC (it must have the same extent
# and resolution of the SOC layer)
# SOC_min<-(1-(SOC_UNC/100))*SOC
# SOC_max<-(1+(SOC_UNC/100))*SOC Define
# SOC min, max Clay min and max.
SOC_min <- SOC * 0.8
SOC_max <- SOC * 1.2
clay_min <- clay * 0.9
clay_max <- clay * 1.1
```

The script then uses the RothC function with the parameters listed below. We want to estimate the annual Carbon inputs needed to reach the actual GSOCmap value (equilibrium C inputs). We will first run the model assuming "Cinputs" equal 1. We can assume that SOC stock values when there are no C inputs ("Cinputs" equal to 0) will be equal to the inert organic carbon stocks (SOC=FallIOM). We can then build a simple linear model to estimate the "Cinputs" value needed to reach SOC FAO actual value (equilibrium C inputs, See modeling approach, Chapter 5 and 6).

```
b <- 1
# C input equilibrium. (Ceq)
fb <- Roth_C(Cinputs = b, years = years,
    DPMptf = 0, RPMptf = 0, BIOptf = 0, HUMptf = 0,
    FallIOM = FallIOM, Temp = Temp, Precip = Precip,
    Evp = Evp, Cov = Cov, Cov1 = Cov1, Cov2 = Cov2,
    soil.thick = soil.thick, SOC = SOC, clay = clay,
    DR = DR, bare1 = bare1)
fb_t <- fb[1] + fb[2] + fb[3] + fb[4] + fb[5]
m <- (fb_t - FallIOM)/(b)
Ceq <- (SOC - FallIOM)/m
```

We will repeat the "C input eq. code" to calculate minimum and maximum carbon inputs at equilibrium, using the combination of environmental variables listed in chapter 12 (Uncertainties).

```
# UNCERTAINTIES C input equilibrium
# (MINIMUM)
FallIOM_min = 0.049 * SOC_min^(1.139)
fb_min <- Roth_C(Cinputs = b, years = years,
    DPMptf = 0, RPMptf = 0, BIOptf = 0, HUMptf = 0,
    FallIOM = FallIOM, Temp = Temp * 1.02,
    Precip = Precip * 0.95, Evp = Evp, Cov = Cov,
    Cov1 = Cov1, Cov2 = Cov2, soil.thick = soil.thick,
    SOC = SOC_min, clay = clay_min, DR = DR,
    bare1 = bare1)
fb_t_MIN <- fb_min[1] + fb_min[2] + fb_min[3] +
    fb_min[4] + fb_min[5]
m <- (fb_t_MIN - FallIOM_min)/(b)
Ceq_MIN <- (SOC_min - FallIOM_min)/m
# UNCERTAINTIES C input equilibrium
# (MAXIMUM)
FallIOM_max = 0.049 * SOC_max^(1.139)
fb_max <- Roth_C(Cinputs = b, years = years,
    DPMptf = 0, RPMptf = 0, BIOptf = 0, HUMptf = 0,
    FallIOM = FallIOM, Temp = Temp * 0.98,
    Precip = Precip * 1.05, Evp = Evp, Cov = Cov,
    Cov1 = Cov1, Cov2 = Cov2, soil.thick = soil.thick,
```

```
    SOC = SOC_max, clay = clay_max, DR = DR,
    bare1 = bare1)
fb_t_MAX <- fb_max[1] + fb_max[2] + fb_max[3] +
    fb_max[4] + fb_max[5]
m <- (fb_t_MAX - FallIOM_max)/(b)
Ceq_MAX <- (SOC_max - FallIOM_max)/m
```

Now for each land use, we will run the "pedotransfer functions" (Weiherm&uuller et al., 2013) to estimate the values of the SOC pools. Here is the example for the croplands land use. Then the same code will be executed for the rest of the land use classes.

```
# SOC POOLS AFTER 500 YEARS RUN WITH C INPUT EQUILIBRIUM
if (LU==2){
RPM_p_2<-((0.184*SOC + 0.1555)*(clay +
                    1.275)^(-0.1158))*0.9902+0.4788
BIO_p_2<-((0.014*SOC + 0.0075)*(clay +
                    8.8473)^(0.0567))*1.09038+0.04055
HUM_p_2<-((0.7148*SOC + 0.5069)*(clay +
                    0.3421)^(0.0184))*0.9878-0.3818
DPM_p_2<-SOC-FallIOM-RPM_p_2-HUM_p_2-BIO_p_2
feq_t<-RPM_p_2+BIO_p_2+HUM_p_2+DPM_p_2+FallIOM
#uncertainties  MIN
RPM_p_2_min<-((0.184*SOC_min + 0.1555)*(clay_min +
                    1.275)^(-0.1158))*0.9902+0.4788
BIO_p_2_min<-((0.014*SOC_min + 0.0075)*(clay_min +
                    8.8473)^(0.0567))*1.09038+0.04055
HUM_p_2_min<-((0.7148*SOC_min + 0.5069)*(clay_min +
                    0.3421)^(0.0184))*0.9878-0.3818
DPM_p_2_min<-SOC_min-FallIOM_min-
  RPM_p_2_min-HUM_p_2_min-BIO_p_2_min
feq_t_min<-RPM_p_2_min+BIO_p_2_min+
  HUM_p_2_min+DPM_p_2_min+FallIOM_min
#uncertainties  MAX
RPM_p_2_max<-((0.184*SOC_max + 0.1555)*(clay_max +
        1.275)^(-0.1158))*0.9902+0.4788
BIO_p_2_max<-((0.014*SOC_max + 0.0075)*(clay_max +
        8.8473)^(0.0567))*1.09038+0.04055
```

```
HUM_p_2_max<-((0.7148*SOC_max + 0.5069)*(clay_max +
        0.3421)^(0.0184))*0.9878-0.3818
DPM_p_2_max<-SOC_max-FallIOM_max-
  RPM_p_2_max-HUM_p_2_max-BIO_p_2_max
feq_t_max<-RPM_p_2_max+BIO_p_2_max+
  HUM_p_2_max+DPM_p_2_max+FallIOM_max
```

Finally, we will save the outputs pools to the variables C_INPUT_EQ :

```
C_INPUT_EQ[i,2]<-SOC
C_INPUT_EQ[i,3]<-Ceq
C_INPUT_EQ[i,4]<-feq_t
C_INPUT_EQ[i,5]<-DPM_p_2
C_INPUT_EQ[i,6]<-RPM_p_2
C_INPUT_EQ[i,7]<-BIO_p_2
C_INPUT_EQ[i,8]<-HUM_p_2
C_INPUT_EQ[i,9]<-FallIOM
C_INPUT_EQ[i,10]<-Ceq_MIN
C_INPUT_EQ[i,11]<-Ceq_MAX
C_INPUT_EQ[i,12]<-feq_t_min
C_INPUT_EQ[i,13]<-DPM_p_2_min
C_INPUT_EQ[i,14]<-RPM_p_2_min
C_INPUT_EQ[i,15]<-BIO_p_2_min
C_INPUT_EQ[i,16]<-HUM_p_2_min
C_INPUT_EQ[i,17]<-FallIOM_min
C_INPUT_EQ[i,18]<-feq_t_max
C_INPUT_EQ[i,19]<-DPM_p_2_max
C_INPUT_EQ[i,20]<-RPM_p_2_max
C_INPUT_EQ[i,21]<-BIO_p_2_max
C_INPUT_EQ[i,22]<-HUM_p_2_max
C_INPUT_EQ[i,23]<-FallIOM_max
}
}
```

In order to properly save the output vector, we will change the names of the fields:

```
# rename de columns
colnames(C_INPUT_EQ@data)[2] = "SOC_FAO"
colnames(C_INPUT_EQ@data)[3] = "Cinput_EQ"
colnames(C_INPUT_EQ@data)[4] = "SOC_pedotransfer"
colnames(C_INPUT_EQ@data)[5] = "DPM_pedotransfer"
colnames(C_INPUT_EQ@data)[6] = "RPM_pedotransfer"
colnames(C_INPUT_EQ@data)[7] = "BIO_pedotransfer"
colnames(C_INPUT_EQ@data)[8] = "HUM_pedotransfer"
colnames(C_INPUT_EQ@data)[9] = "IOM_pedotransfer"
colnames(C_INPUT_EQ@data)[10] = "CIneq_min"
colnames(C_INPUT_EQ@data)[11] = "CIneq_max"
colnames(C_INPUT_EQ@data)[12] = "SOC_min"
colnames(C_INPUT_EQ@data)[13] = "DPM_min"
colnames(C_INPUT_EQ@data)[14] = "RPM_min"
colnames(C_INPUT_EQ@data)[15] = "BIO_min"
colnames(C_INPUT_EQ@data)[16] = "HUM_min"
colnames(C_INPUT_EQ@data)[17] = "IOM_min"
colnames(C_INPUT_EQ@data)[18] = "SOC_max"
colnames(C_INPUT_EQ@data)[19] = "DPM_max"
colnames(C_INPUT_EQ@data)[20] = "RPM_max"
colnames(C_INPUT_EQ@data)[21] = "BIO_max"
colnames(C_INPUT_EQ@data)[22] = "HUM_max"
colnames(C_INPUT_EQ@data)[23] = "IOM_max"
```

Finally, we will set the output directory and save the output vector:

```
# SAVE the Points (shapefile)
setwd(WD_FOLDER)
writeOGR(C_INPUT_EQ, ".", "OUTPUTS/1_SPIN_UP/SPIN_UP_County_AOI",
    driver = "ESRI Shapefile")
```

10.2.2 Script Number 13.B. "ROTH_C_SPIN_UP_UNC_v66.R" (analytical solution)

Script number 13.B implements the first modeling phase (spin up) using the analytical solution approach (see section 5.4.1.2), developed by Dechow et al. (2019) and adapted to spatial simulations.

```
# 12/11/2020

# SPATIAL SOIL R for VECTORS

###### SPIN UP ################

# MSc Ing Agr Luciano E Di Paolo Dr Ing
# Agr Guillermo E Peralta Dr. Ing Rene
# Dechow

# 13_1_ROTHC_C_SPIN_UP_UNC_v66.R This
# script does some regionalized
# uncertainty runs with RothC it
# quantifies Pool distributions and
# equilibrium C input for a minimum and
# maximum scenario Input: a point.shp
# file with SOC and un

# SOilR from Sierra, C.A., M. Mueller,
# S.E. Trumbore (2012).  Models of soil
# organic matter decomposition: the
# SoilR package, version 1.0
# Geoscientific Model Development,
# 5(4), 1045-1060.
# http://www.geosci-model-dev.net/5/1045/2012/gmd-5-1045-2012.html.
```

In this script we will load the stack generated in script number 10 and the target points (Section 9.7, QGIS model number 1). We will obtain an output vector containing our target points. This script estimates the SOC stocks for the different pools and the equilibrium carbon inputs (the carbon inputs needed to reach the 2001 SOC stocks),assuming homogeneous soil, climatic and management conditions. First, the script estimates the fractions (fi) of DPM, RPM, BIO and HUM pools at equilibrium following the set of equations and intermediate coefficients described in the supplementary material of Dechow et el. (2019). Once the fractions of the different pools are estimated, the amount of Carbon (tC ha^{-1}) in each pool is estimated from the total SOC stock. Finally, Carbon inputs (Ci) at equilibrium are estimated from the generated results. All results will be saved to the output vector (shapefile file). Using this approach, the spin

up phase will allow us to obtain two key outputs that will be saved to a point vector layer called "C_INPUT_EQ.shp": equilibrium carbon inputs (Ceq) and the carbon stocks of the different soil C pools to run the second phase (warm up phase). First, we need to load the packages into R:

```
rm(list=ls())

library(SoilR)
library(raster)
library(rgdal)
library(soilassessment)
```

Then we define the function to estimate the Inert Organic Carbon content (IOM; t C ha^{-1}) according to the equation (see eq. 5.2) given by Falloon et al. (1998):

```
# calculates some iom in t / ha
# input
# 1. c total carbpn stock in t /ha

fIOM.Falloon.RothC =function(c, par1=-1.31, par2=1.139)
{

 # IOM=10^(par1+par2*log10(c))
  IOM=0.049*SOC^(1.139)
  IOM
}
```

Then from lines 61-176 we will define the main function that will estimate the fraction of each carbon pool at equilibrium and the carbon inputs at equilibrium ("fget_equilibrium_factions.RothC_input").

The inputs of this function are (line 61):

- xi = is a scalar representing an averaged rate modifying factor (rmf; average of temperature, soil moisture, vegetation cover and anaerobic/paddy rice factors).

- *C.tot* = represents the initial C stock (and therefore C stock in equilibrium; that will correspond to the FAO GSOCmap)

- *clay* = clay content in percent
- *fractI* = vector of Cinput fractions (these fractions 1, 2 and 3 correspond to the γ_{DPM}, γ_{RPM}, γ_{HUM} partition coefficients, that represent the proportion of DPM, RPM and HUM of the incoming residues. The fractions are derived from the decomposability of incoming C inputs (e.g. DPM/RPM ratio). The way to estimate these γ_{DPM}, γ_{RPM}, γ_{HUM} partition coefficients (named as "fractI" in the script) is going to be defined later at line 229.

```
#
# get_equilibrium_fractions.RothC_input
# brief: quantifies pool distribution and
#C input for RothC at equilibrium
#Input
# xi= scalar representing an averaged modifying factor
# C.tot = initial C stock (and C stock in equilibrium)
# clay = clay content
# fractI = vector of Cinput fractions
#that enter the DPM, RPM, HUM
# with a DR of 1.44 fractI becomes
# [1] 0.5901639 0.4098361 0.0000000
# by fractI=c((DR)/(DR+1),1-(DR)/(DR+1),0)
#Output
# list with pools at equilibrium and C input at equilibrium
#

fget_equilibrium_fractions.RothC_input=function(xi=1,
            C.tot,clay, fractI)
{
  rmf=xi
  IOM= fIOM.Falloon.RothC(c = C.tot)
  C.active=C.tot-IOM
```

The output of this function (at line 174) will be a list of two elements: the first element is a vector of 5 elements containing the C stocks of the different SOC pools and the second element is the carbon input at equilibrium.

From lines 67-175, the analytical solution to estimate SOC pools and estimate equilibrium Carbon inputs is implemented. First we will need to define the proportion of decomposed carbon that goes to the BIO pool (46 percent; "fract.rooted.to.bio") or to the HUM pool (54 percent; "fract.rooted.to.hum") using the default coefficients from the original RothC model. At lines 76-82 we will define the decomposition rates (k) for each pool (using the default k values of the original RothC model).

```
#The analytical solution of RothC
# Parameter
fract.rooted.to.bio = 0.46
fract.rooted.to.hum = 0.54
ks = c(k.DPM = 10, k.RPM = 0.3, k.BIO = 0.66,
       k.HUM = 0.02, k.IOM = 0)
ks=as.numeric(ks)
k.dpm=ks[1]
k.rpm=ks[2]
k.bio=ks[3]
k.hum=ks[4]
```

Then we will define the Carbon Use Efficiency (CUE) at line 86. We will require the CUE later to estimate intermediate coefficients (α). The CUE represents the amount of carbon that goes to the BIO + HUM pools (from the total decomposed carbon). The amount of carbon that is outputted in form of CO2 or stored in the BIO and HUM pools is determined by the clay content of the soil (following the original equation in the RothC model), so the CUE is dependent on clay content.

```
# the carbon use efficiency
cue=  1/(1+ 1.67 * (1.85 + 1.6 * exp(-0.0786 * clay)))
```

From lines 88-113, we will need to define a number of intermediate coefficients (referred as α, λ, and c in the equations detailed in the supplementary material of Dechow et el. (2019)). These coefficients will then allow us to estimate the fraction of each carbon pool at equilibrium. α 1 and α 2 coefficients will be estimated from the carbon use efficiency and from the proportion of carbon that goes to BIO or HUM (defined in lines 74-75). The intermediate coefficients α 1.1 to α 2.2 will be derived from the decomposition rates of each pool already defined

in lines 76-82; and from the average rate modifying factor (rmf; average of temperature, soil moisture, vegetation cover and anaerobic/paddy rice factors). The c coefficient will be estimated from the previously estimated α coefficients (lines 111-113). λ (Lambda) intermediate coefficients (lines 106-107) are not required in this stationary version and are currently disabled.

```
# All the coefficients alpha.1 und
# alpha.2

alpha.1 = cue * fract.rooted.to.bio
alpha.2 = cue * fract.rooted.to.hum


# All the coefficients a.1.1, a.1.2,
# a.2.1, a2.2

a.1.1 = k.bio * rmf * (alpha.1 - 1)
a.1.2 = alpha.1 * k.hum * rmf
a.2.1 = alpha.2 * k.bio * rmf
a.2.2 = k.hum * rmf * (alpha.2 - 1)


# The Eigenvalues lambda 1 and lambda 2

lambda.1 = (a.1.1 + a.2.2)/2 - sqrt(((a.1.1 +
    a.2.2)/2) * ((a.1.1 + a.2.2)/2) + a.1.2 *
    a.2.1 - a.1.1 * a.2.2)
lambda.2 = (a.1.1 + a.2.2)/2 + sqrt(((a.1.1 +
    a.2.2)/2) * ((a.1.1 + a.2.2)/2) + a.1.2 *
    a.2.1 - a.1.1 * a.2.2)

# The c.0.1; c.0.2; c.0.3 values

c.0.1 = (alpha.2 * a.1.2 - alpha.1 * a.2.2)/(a.1.1 *
    a.2.2 - a.1.2 * a.2.1)
c.0.2 = (alpha.2 * a.1.2 - alpha.1 * a.2.2)/(a.1.1 *
    a.2.2 - a.1.2 * a.2.1)
c.0.3 = (a.1.2)/(a.1.1 * a.2.2 - a.1.2 *
    a.2.1)
```

Then, from lines 115-155, we will define the u coefficients (see intermediate equations in the supplementary material of Dechow et el. (2019)), which are functions integrating model structure and related to the proportion of C of each pool entering other pools. These u coefficients will be estimated from the previously defined α and c coefficients, from the decomposition rates (k) and from the rate modifying factor (rmf). We will define in total a set of 11 u coefficients, that we will finally require to estimate the SOC fraction of each pool.

```
# BIO pool quantification

u.bio.dpm=(c.0.2) #65
u.bio.rpm=(c.0.1) #66
u.bio.hum=(c.0.3) #67

# HUM pool quantification ( is all C.78)

u.hum.dpm= 1/a.1.2*((-c.0.2*a.1.1-alpha.1))
u.hum.rpm= 1/a.1.2*(-c.0.2*a.1.1-alpha.1)
u.hum.hum= 1/a.1.2*(-c.0.3*a.1.1)

# DPM C ( is all C.79)

u.dpm.dpm=1/k.dpm/rmf

#C.dpm=i.dpm * u.dpm.dpm + C0 * s.dpm

# RPM C ( is all C.80)

u.rpm.rpm=1/k.rpm/rmf

#C.rpm=i.rpm * u.rpm.rpm + C0 *s.rpm

# Total C ( is all C.78)

u.dpm=u.dpm.dpm+u.bio.dpm+u.hum.dpm
u.rpm=u.rpm.rpm+u.bio.rpm+u.hum.rpm
u.hum=u.bio.hum+u.hum.hum
```

To end the "fget_equilibrium_factions.RothC_input" function, we will define how to estimate the fraction (fi) of each active SOC pool to the total SOC at equilibrium. These fractions are going to be estimated following the equations 5.5 to 5.8 already explained in section 5.4.1.2, from the u coefficients and the γ_{DPM}, γ_{RPM}, γ_{HUM} partition coefficients (fract I):

$$f_{DPM} = \frac{DPM\ u_{DPM}}{DPM\ u_{DPM} + RPM\ u_{RPM} + HUM\ u_{HUM}} \tag{5.5}$$

$$f_{RPM} = \frac{RPM\ u_{RPM}}{DPM\ u_{DPM} + RPM\ u_{RPM} + HUM\ u_{HUM}} \tag{5.6}$$

$$f_{BIO} = \frac{DPM\ u_{BIO}DPM + RPM\ u_{BIO}RPM + HUM\ u_{BIO}HUM}{DPM\ u_{DPM} + RPM\ u_{RPM} + HUM\ u_{HUM}} \tag{5.7}$$

$$f_{HUM} = \frac{\gamma_{DPM}\ u_{HUM}DPM + RPM\ u_{HUM}RPM + HUM\ u_{HUM}HUM}{DPM\ u_{DPM} + RPM\ u_{RPM} + HUMu_{HUM}} \tag{5.8}$$

As all pool fractions will be estimated by dividing by the same term ($\gamma_{DPM}u_{DPM} + RPM\ u_{RPM} + HUM\ u_{HUM}$) we will define it in line 157. Then we will define the equations to estimate the fraction of each pool in lines 159-162. Finally, at line 164 we will create a vector containing the fraction of each active pool (f_{DPM}, d_{RPM}, f_{BIO}, f_{HUM}).

```
denominator = fractI[1] * u.dpm + fractI[2] *
    u.rpm + fractI[3] * u.hum

fract.dpm = fractI[1] * u.dpm.dpm/denominator
fract.rpm = fractI[2] * u.rpm.rpm/denominator
fract.bio = (fractI[1] * u.bio.dpm + fractI[2] *
    u.bio.rpm + fractI[3] * u.bio.hum)/denominator
fract.hum = (fractI[1] * u.hum.dpm + fractI[2] *
    u.hum.rpm + fractI[3] * u.hum.hum)/denominator

fract.all = c(fract.dpm, fract.rpm, fract.bio,
    fract.hum)
```

The SOC stocks (t C/ha) of each pool will be estimated by multiplying the fraction of each active pool by the total active SOC stock (t C/ha) (line 169). We will define a vector containing all four active fractions (line 170), plus the IOM fraction (line 171) (total 5 fractions); and a vector containing the SOC stocks of all five pools (line 172).

```
  # IOM

  fract.all_stock=(fract.all*C.active)
  fract.all=fract.all_stock/C.tot
  fract.all=append(fract.all,IOM/C.tot)
  pools=fract.all*C.tot
```

In the last step of this function, we will define how to estimate C inputs following equation 5.11 explained in section 5.4.1.2 (line 173); and we will define the output of this function as a list containing the C inputs at equilibrium plus the SOC stocks of all five pools of the RothC model (line 174).

```
Cin=(C.tot-pools[5])/denominator
  list(pools,Cin)
}
```

From lines 180-217, we will set the working directory, and create a vector that contains the results and extract the data from the already created spin up stack (total SOC stocks, clay content, DPR/RPM ratio, Land use class)

```
# Set working directory

WD_FOLDER=("D:/TRAINING_MATERIALS")

# Vector must be an empty points vector.

setwd(WD_FOLDER)
Vector<-readOGR(
  "INPUTS/TARGET_POINTS/target_points.shp")
#Vector2<-readOGR("INPUTS/TARGET_POINTS/
#   Deutschland/target_points.shp")
# Stack_Set_1 contains the spatial variables
```

```
setwd(WD_STACK)
Stack_Set_1<- stack("Stack_Set_SPIN_UP_AOI.tif")

# Create A vector to save the results

C_INPUT_EQ<-Vector

# use this only for backup

# C_INPUT_EQ<-readOGR("OUTPUTS/1_SPIN_UP/
#   SPIN_UP_BSAS_27-03-2020_332376.shp")

# extract variables to points

Vector_variables<-extract(Stack_Set_1,Vector,df=TRUE)
# Extract the layers from the Vector

SOC_im<-Vector_variables[[2]] # first part of stack

clay_im<-Vector_variables[[3]] # second part of stack

DR_im<-Vector_variables[[40]]

LU_im<-Vector_variables[[41]]

# Define Years for Cinputs calculations

#years=seq(1/12,2000,by=1/12)
```

At line 221 , we will set the function called " Roth_C_equi_analy " . This function will be iterated over the target points, calculating the soil carbon pools defined in the Roth C model. The inputs of this function are:

- C inputs (carbon inputs),
- Temp (temperature)
- Precip (precipitation)
- Evp (evapotranspiration)
- Cov2 (vegetation cover) soil.thick (soil thickness = 30 cm)

- SOC (initial total SOC stocks)
- clay (clay content at 0-30cm)
- DR (the DPM/RPM ratio)
- bare1 (coefficient indicating whether the soil is covered or vegetated)
- LU (land use class)

The outputs of this function are two: 1) The five soil carbon pools (result[[1]]), 2) the carbon inputs of equilibrium (result[[2]]). The previous function "fget_equilibrium_factions.RothC_input" is inside this function is . The main purpose of this function is to get the values of the input variables for the "fget_equilibrium_factions.RothC_input" function.

```
# ROTH C MODEL FUNCTION .

########## function set up starts###
Roth_C_equi_analy<-function(Cinputs,Temp,Precip,
          Evp,Cov2,soil.thick,SOC,clay,DR,bare1,LU)

{
```

So , first, at line 228 we calculate a vector of three elements, called "FractI". This variable represents the carbon input fraction of DPM , RPM and HUM (the γ_{DPM}, γ_{RPM}, γ_{HUM} partition coefficients). Normally, if we do not have organic amendments , the humic fraction is set to 0.

```
  #-------------------------
  # c input distribution
  #-------------------------
  fractI=c((DR)/(DR+1),1-(DR)/(DR+1),0)
```

Then from lines 234 to 268 , the rate modifying factors are calculated : Paddy Field coefficient (fPR), Temperature coefficient (fT), Soil Moisture coefficient (fW_2), and Vegetation cover coefficient (fC).

```
# Paddy fields coefficent fPR = 0.4 if
# the target point is class = 13 , else
# fPR=1 From Shirato and Yukozawa 2004
```

```
fPR = (LU == 13) * 0.4 + (LU != 13) * 1

# Temperature effects per month
fT = fT.RothC(Temp[, 2])

# Moisture effects per month .

fw1func <- function(P, E, S.Thick = 30, pClay = 32.0213,
    pE = 1, bare) {

    M = P - E * pE
    Acc.TSMD = NULL
    for (i in 2:length(M)) {
        B = ifelse(bare[i] == FALSE, 1, 1.8)
        Max.TSMD = -(20 + 1.3 * pClay - 0.01 *
            (pClay^2)) * (S.Thick/23) * (1/B)
        Acc.TSMD[1] = ifelse(M[1] > 0, 0,
            M[1])
        if (Acc.TSMD[i - 1] + M[i] < 0) {
            Acc.TSMD[i] = Acc.TSMD[i - 1] +
                M[i]
        } else (Acc.TSMD[i] = 0)
        if (Acc.TSMD[i] <= Max.TSMD) {
            Acc.TSMD[i] = Max.TSMD
        }
    }
    b = ifelse(Acc.TSMD > 0.444 * Max.TSMD,
        1, (0.2 + 0.8 * ((Max.TSMD - Acc.TSMD)/(Max.TSMD -
            0.444 * Max.TSMD))))
    b <- clamp(b, lower = 0.2)
    return(data.frame(b))
}

fW_2 <- fw1func(P = (Precip[, 2]), E = (Evp[,
    2]), S.Thick = soil.thick, pClay = clay,
    pE = 1, bare = bare1)$b

# Vegetation Cover effects
```

```
fC <- Cov2[, 2]
```

After that, the "xi" variable is calculated as the mean of the product of the four modifying factors.

```
  # Set the factors frame for Model calculations

  xi=mean(fT*fW_2*fC*fPR)
```

Finally, we calculate the "result" variable using the "fget_equilibrium_fractions.\ RothC_input" function. We have already calculated or set the input variables for that function. The output of the function " Roth_C_equi_analy " is a two element list containing both the soil carbon pools and the carbon input of equilibrium.

```
# RUN THE MODEL

  result=fget_equilibrium_fractions.RothC_input(xi=xi,
                  C.tot=SOC,clay=clay, fractI)

  # RUN THE MODEL FROM SOILR
  #Model3_spin=RothCModel(t=years,
  #   C0=c(DPMptf, RPMptf, BIOptf, HUMptf, FallIOM),
  #   In=Cinputs,DR=DR,clay=clay,xi=xi.frame, pass=TRUE)
  #Ct3_spin=getC(Model3_spin)

  # Get the final pools of the time series


  return(result)
}
######### function set up ends###############
```

After setting the " Roth_C_equi_analy " function we can now iterate that function over our target points (line 293) .

```
# Iterates over the area of interest
#source("D:/projecte/Rlibs/dataframe_ops.R")

########for loop starts###############3
for (i in 1:dim(Vector_variables)[1]) {
```

From lines 297-310 we will extract the climate and vegetation cover variables. At line 297 we will create a vector called "Vect". This variable is a vector of 53 elements containing all the variables we need to run the model over a specific target point. At lines 299-300 we extract from the "Vect" variable the temperature information for the ith target point. At lines 302-303 we extract from the "Vect" variable the precipitation information for the ith target point. At lines 305-306 we extract from the "Vect" variable the potential evapotranspiration information for the ith target point. At lines 308-310 we extract from the "Vect" variable the vegetation cover information for the ith target point.

```
# Extract the variables

Vect <- as.data.frame(Vector_variables[i,
    ])

Temp <- as.data.frame(t(Vect[4:15]))
Temp <- data.frame(Month = 1:12, Temp = Temp[,
    1])

Precip <- as.data.frame(t(Vect[16:27]))
Precip <- data.frame(Month = 1:12, Precip = Precip[,
    1])

Evp <- as.data.frame(t(Vect[28:39]))
Evp <- data.frame(Month = 1:12, Evp = Evp[,
    1])

Cov <- as.data.frame(t(Vect[42:53]))
Cov1 <- data.frame(Cov = Cov[, 1])
Cov2 <- data.frame(Month = 1:12, Cov = Cov[,
    1])
```

To avoid calculus over points with null or odd inputs values, we will introduce and "if" statement at line 314:

```
 #Avoid calculus over Na values

  if (any(is.na(Evp[,2])) | any(is.na(Temp[,2])) |
      any(is.na(SOC_im[i])) | any(is.na(clay_im[i])) |
      any(is.na(Precip[,2]))  |  any(is.na(Cov2[,2]))  |
      any(is.na(Cov1[,1]))  | any(is.na(DR_im[i])) |
      (SOC_im[i]<0) | (clay_im[i]<0) ) {C_INPUT_EQ[i,2]<-0
  }else{
```

From line 321 to line 327 we will set the environmental variables for the ith target point.

```
# Set the variables from the images

soil.thick = 30  #Soil thickness (org. layer topsoil)
SOC <- SOC_im[i]  #Soil organic carbon in Mg/ha
clay <- clay_im[i]  #Percent clay %

DR <- DR_im[i]  # DPM/RPM (decomposable vs
# resistant plant material.)
bare1 <- (Cov1 > 0.8)  # If surface is bare or vegetated
LU <- LU_im[i]
```

At line 330 we will calculate the inert organic carbon with the Falloon method.

```
    #IOM using Falloon method
    FallIOM=0.049*SOC^(1.139)
```

From line 339 to line 342 we will define the maximum and minimum values for the confidence interval for the variables SOC and Clay, in order to estimate the uncertainties.

```
# If you use a SOC uncertainty layer
# turn on this. First open the layer
# SOC_UNC (it must have the same extent
```

```
# and resolution of the SOC layer)

# SOC_min<-(1-(SOC_UNC/100))*SOC
# SOC_max<-(1+(SOC_UNC/100))*SOC

# Define SOC min, max Clay min and max.
SOC_min <- SOC * 0.8
SOC_max <- SOC * 1.2
clay_min <- clay * 0.9
clay_max <- clay * 1.1
```

At line 358 we will calculate the "result" object by using the "Roth_C_equi_analy" function. The "result" object is a list of two elements, the first one is a vector of five elements representing the five soil carbon pools, the second element is the carbon input of equilibrium. Then, at line 364 and 370 we will apply again the "Roth_C_equi_analy" function for the minimum and maximum inputs values, to get the uncertainties.

```
# C input equilibrium. (Ceq) + Ceq_MIN
# + Ceq_MAX are quantified here

# fb<-Roth_C(Cinputs=b,years=years,DPMptf=0,
# RPMptf=0, BIOptf=0, HUMptf=0,
# FallIOM=FallIOM,Temp=Temp,Precip=Precip,Evp=Evp,
# Cov=Cov,Cov1=Cov1,Cov2=Cov2,soil.thick=soil.thick,
# SOC=SOC,clay=clay,DR=DR,bare1=bare1,LU=LU)
# fb_t<-fb[1]+fb[2]+fb[3]+fb[4]+fb[5]
# pool.equi.goodi[i,]=fb
# m<-(fb_t-FallIOM)/(b)

# Ceq<-(SOC-FallIOM)/m

# Cin.equi$spinup[i]=Ceq;

result = Roth_C_equi_analy(Cinputs = b, Temp = Temp,
    Precip = Precip, Evp = Evp, Cov2 = Cov2,
    soil.thick, SOC, clay, DR, bare1, LU)
Ceq = result[[2]]
```

```
pool.equi.mean = result[[1]]

# UNCERTAINTIES C input equilibrium
# (MINIMUM)

result = Roth_C_equi_analy(Cinputs = b, Temp = Temp,
    Precip = Precip, Evp = Evp, Cov2 = Cov2,
    soil.thick, SOC_min, clay_min, DR, bare1,
    LU)
Ceq_MIN = result[[2]]
pool.equi.min = result[[1]]
```

From line 374 to line 430 we will save all the results in the vector called "C_INPUT_EQ".

```
 # UNCERTAINTIES C input equilibrium (MAXIMUM)

    result=Roth_C_equi_analy(Cinputs=b,Temp=Temp,Precip=Precip,
      Evp=Evp,Cov2=Cov2,soil.thick,SOC_max,clay_max,DR,bare1,LU)
    Ceq_MAX = result[[2]]
    pool.equi.max = result[[1]]

    # SOC POOLS AFTER 500 YEARS RUN WITH C INPUT EQUILIBRIUM
    good_landuse_classes=c(2,12,13,4,3,5,6,8)
    if (LU %in% good_landuse_classes){

      C_INPUT_EQ[i,2]<-SOC
      C_INPUT_EQ[i,3]<-Ceq
      C_INPUT_EQ[i,4]<-sum(pool.equi.mean)
      C_INPUT_EQ[i,5]<-pool.equi.mean[1] #DPM
      C_INPUT_EQ[i,6]<-pool.equi.mean[2] #RPM
      C_INPUT_EQ[i,7]<-pool.equi.mean[3] #BIO
      C_INPUT_EQ[i,8]<-pool.equi.mean[4] #HUM
      C_INPUT_EQ[i,9]<-pool.equi.mean[5] #IOM
      C_INPUT_EQ[i,10]<-Ceq_MIN
      C_INPUT_EQ[i,11]<-Ceq_MAX
      C_INPUT_EQ[i,12]<-sum(pool.equi.min)
      C_INPUT_EQ[i,13]<-pool.equi.min[1] #DPM
```

```
    C_INPUT_EQ[i,14]<-pool.equi.min[2] #RPM
    C_INPUT_EQ[i,15]<-pool.equi.min[3] #BIO
    C_INPUT_EQ[i,16]<-pool.equi.min[4] #HUM
    C_INPUT_EQ[i,17]<-pool.equi.min[5] #IOM
    C_INPUT_EQ[i,18]<-sum(pool.equi.max)
    C_INPUT_EQ[i,19]<-pool.equi.max[1] #DPM
    C_INPUT_EQ[i,20]<-pool.equi.max[2] #RPM
    C_INPUT_EQ[i,21]<-pool.equi.max[3] #BIO
    C_INPUT_EQ[i,22]<-pool.equi.max[4] #HUM
    C_INPUT_EQ[i,23]<-pool.equi.max[5] #IOM

  }else {
    C_INPUT_EQ[i,2]<-SOC
    C_INPUT_EQ[i,3]<-Ceq
    C_INPUT_EQ[i,4]<-0
    C_INPUT_EQ[i,5]<-0
    C_INPUT_EQ[i,6]<-0
    C_INPUT_EQ[i,7]<-0
    C_INPUT_EQ[i,8]<-0
    C_INPUT_EQ[i,9]<-0
    C_INPUT_EQ[i,10]<-0
    C_INPUT_EQ[i,11]<-0
    C_INPUT_EQ[i,12]<-0
    C_INPUT_EQ[i,13]<-0
    C_INPUT_EQ[i,14]<-0
    C_INPUT_EQ[i,15]<-0
    C_INPUT_EQ[i,16]<-0
    C_INPUT_EQ[i,17]<-0
    C_INPUT_EQ[i,18]<-0
    C_INPUT_EQ[i,19]<-0
    C_INPUT_EQ[i,20]<-0
    C_INPUT_EQ[i,21]<-0
    C_INPUT_EQ[i,22]<-0
    C_INPUT_EQ[i,23]<-0

  }
  print(c(i,SOC,Ceq))
```

```
  } # NA problems
}
################for loop ends###############
```

From line 434 to line 455 we will change the names of the columns of the vector files "C_INPUT_EQ" to the variables we want to obtain.

```
#rename the columns

colnames(C_INPUT_EQ@data)[2]="SOC_FAO"
colnames(C_INPUT_EQ@data)[3]="Cinput_EQ"
colnames(C_INPUT_EQ@data)[4]="SOC_pedotransfer"
colnames(C_INPUT_EQ@data)[5]="DPM_pedotransfer"
colnames(C_INPUT_EQ@data)[6]="RPM_pedotransfer"
colnames(C_INPUT_EQ@data)[7]="BIO_pedotransfer"
colnames(C_INPUT_EQ@data)[8]="HUM_pedotransfer"
colnames(C_INPUT_EQ@data)[9]="IOM_pedotransfer"
colnames(C_INPUT_EQ@data)[10]="CIneq_min"
colnames(C_INPUT_EQ@data)[11]="CIneq_max"
colnames(C_INPUT_EQ@data)[12]="SOC_min"
colnames(C_INPUT_EQ@data)[13]="DPM_min"
colnames(C_INPUT_EQ@data)[14]="RPM_min"
colnames(C_INPUT_EQ@data)[15]="BIO_min"
colnames(C_INPUT_EQ@data)[16]="HUM_min"
colnames(C_INPUT_EQ@data)[17]="IOM_min"
colnames(C_INPUT_EQ@data)[18]="SOC_max"
colnames(C_INPUT_EQ@data)[19]="DPM_max"
colnames(C_INPUT_EQ@data)[20]="RPM_max"
colnames(C_INPUT_EQ@data)[21]="BIO_max"
colnames(C_INPUT_EQ@data)[22]="HUM_max"
colnames(C_INPUT_EQ@data)[23]="IOM_max"
```

Finally, at line 459 we will set the working directory where we are going to savethe output file, and at line 460 we save the vector to an ESRI shapefile file.

```
# SAVE the Points (shapefile)
```

```
setwd("C:/TRAINING_MATERIALS/OUTPUTS/1_SPIN_UP")
writeOGR(C_INPUT_EQ, ".", "SPIN_UP_Country_AOI",
    driver = "ESRI Shapefile", overwrite = TRUE)
```

10.3 Warm up phase

10.3.1 Script Number 14. "ROTH_C_WARM_UP_UNC_v4_parallel.R"No Land use change

Script number 14 implements the second modeling phase ("Warm up" phase). In this script we will load the stack of different layers generated in script number 11 and the target points. We also will load the output vector of the phase 1 (spin up), the climate layers from script number 2, the NPP layer from script number 5, and the land use layer stack from script number 9. This script runs the Roth C model for 18 years (2000-2018) with the possibility to be modified to 20 years if data is available (2000-2020). The final outputs are SOC stocks of the five C pools of the RothC model (DPM, RPM, BIO, HUM and IOM), and the total SOC stock. This information will be saved to a shapefile vector.

RothC warm-up.

Input	Output
Point vector with the locations to run the model (from QGIS procedure 1): (empty vector, should come from the SOCmap FAO, one point per pixel) C_INPUT_EQ.shp (from script number 13)	WARM_UP.shp (contains the output of the model from 2000 to 2018)
Stack layer (from script number 12): NPP_MIAMI_MEAN_81-00_[country_code].tif NPP_MIAMI_MEAN_81-00_MIN[country_code].tif NPP_MIAMI_MEAN_81-00_MAX[country_code].tif TerraClimate layers (from script number 2):	
Prec_Stack_216_01-18_TC.tif Prec_Stack_216_01-18_TC.tif Prec_Stack_216_01-18_TC.tif	

This script runs the spatial RothC model for the warm-up period (from 2001 to 2018). We will provide the script the target points (empty vector layer from Qgis procedure number 1), the Stack layer (from script number 11), the three NPP layers (from script number 5) and the three climate layers generated in script number 2. The output vector layer from script number 13 (Spin up phase) will also be needed.

```
# ROTH C phase 3: WARM UP
#
# MSc Ing Agr Luciano E Di Paolo
# Dr Ing Agr Guillermo E Peralta
###################################
#SoilR from Sierra, C.A., M. Mueller, S.E. Trumbore (2012).
#Models of soil organic matter decomposition: the SoilR package,
#version 1.0 Geoscientific Model Development, 5(4),
#1045--1060.
#http://www.geosci-model-dev.net/5/1045/2012/gmd-5-1045-2012.html.
#####################################

#Empty global environment
rm(list=ls())

#Load packages
library(SoilR)
library(raster)
library(rgdal)
library(soilassessment)
library(foreach)
library(doParallel)


#Define a path to the working directory
wd <-"C:/TRAINING_MATERIALS"
#Set the working directory
setwd(wd)

#Open empty vector
Vector<-readOGR(
  "INPUTS/TARGET_POINTS/target_points_sub.shp")
```

```
#Open Warm Up Stack
Stack_Set_warmup<- stack(
  "INPUTS/STACK/Stack_Set_WARM_UP_AOI.tif")

# Open Result from the spin up phase
#A vector with 5 columns, one for each pool
Spin_up<-readOGR(
  "OUTPUTS/1_SPIN_UP/SPIN_UP_Country_AOI.shp")


# Open Precipitation , temperature, and Evapotranspiration file
# TERRA CLIMATE LAYERS .
#Use the layers downloaded from GEE and a scaling factors of 0.1
setwd(wd)
PREC<-stack(
  "INPUTS/TERRA_CLIMATE/Precipitation_2001-2018.tif")
TEMP<-stack(
  "INPUTS/TERRA_CLIMATE/AverageTemperature_2001-2018.tif")*0.1
PET<-stack(
  "INPUTS/TERRA_CLIMATE/PET_2001-2018.tif")*0.1

#Also check that the number of layers of DR
# and Land use are the same as the years of climate data.

#Open Mean NPP MIAMI 1981 - 2000
NPP<-raster("INPUTS/NPP/NPP_MIAMI_MEAN_81-00_AOI.tif")

NPP_MEAN_MIN<-raster(
  "INPUTS/NPP/NPP_MIAMI_MEAN_81-00_AOI_MIN.tif")

NPP_MEAN_MAX<-raster(
  "INPUTS/NPP/NPP_MIAMI_MEAN_81-00_AOI_MAX.tif")

#Open the land use layer (year 2000).
LU_AOI<-raster(
  "INPUTS/LAND_USE/ESA_Land_Cover_12clases_FAO_AOI.tif")
```

```
NPP<-resample(NPP,LU_AOI,method='bilinear')
NPP_MEAN_MIN<-resample(NPP_MEAN_MIN,LU_AOI,method='bilinear')
NPP_MEAN_MAX<-resample(NPP_MEAN_MAX,LU_AOI, method='bilinear')

#Apply NPP coefficients
NPP<-(LU_AOI==2 | LU_AOI==12 | LU_AOI==13)*NPP*0.53+
  (LU_AOI==4)*NPP*0.88 + (LU_AOI==3 | LU_AOI==5 |
                            LU_AOI==6 | LU_AOI==8)*NPP*0.72
NPP_MEAN_MIN<-(LU_AOI==2 | LU_AOI==12 | LU_AOI==13)*
  NPP_MEAN_MIN*0.53+ (LU_AOI==4)*NPP_MEAN_MIN*0.88 +
  (LU_AOI==3 | LU_AOI==5 | LU_AOI==6 | LU_AOI==8)*
  NPP_MEAN_MIN*0.72
NPP_MEAN_MAX<-(LU_AOI==2 | LU_AOI==12 | LU_AOI==13)*
  NPP_MEAN_MAX*0.53+ (LU_AOI==4)*NPP_MEAN_MAX*0.88 +
  (LU_AOI==3 | LU_AOI==5 | LU_AOI==6 | LU_AOI==8)*
  NPP_MEAN_MAX*0.72
```

As we did in the "spin up" script, we will extract all variables to the target points and create an empty variable to save the results of the "warm up" process.

```
# Extract variables to points
Vector_points <- extract(Stack_Set_warmup,
    Vector, sp = TRUE)
Vector_points <- extract(TEMP, Vector_points,
    sp = TRUE)
Vector_points <- extract(PREC, Vector_points,
    sp = TRUE)
Vector_points <- extract(PET, Vector_points,
    sp = TRUE)
Vector_points <- extract(NPP, Vector_points,
    sp = TRUE)
Vector_points <- extract(NPP_MEAN_MIN, Vector_points,
    sp = TRUE)
Vector_points <- extract(NPP_MEAN_MAX, Vector_points,
    sp = TRUE)

WARM_UP <- Vector
```

Now, we will set some variables in order to run the model for the number of years according to those set for the climate layers. In this example we are running 18 years, but it can be run for less or more years.

```
# Warm Up number of years simulation
yearsSimulation <- dim(TEMP)[3]/12

clim_layers <- yearsSimulation * 12

nppBand <- nlayers(Stack_Set_warmup) + clim_layers *
    3 + 2

firstClimLayer <- nlayers(Stack_Set_warmup) +
    2

nppBand_min <- nppBand + 1

nppBand_max <- nppBand + 2

nDR_beg <- (16 + yearsSimulation)
nDR_end <- nDR_beg + (yearsSimulation - 1)

# land use change simulation (optional,
# if data is available) nLU_beg<-16
# nLU_end<-nLU_beg+(yearsSimulation-1)

# Extract the layers from the Vector

# SOC_im<-Vector_points[[2]]

# clay_im<-Vector_points[[3]]

# LU_im<-Vector_points[[16]] with land
# use change simulation
# LU_im<-Vector_points[[16:34]]
# NPP_im<-Vector_points[[nppBand]]

# NPP_im_MIN<-Vector_points[[nppBand_min]]
```

```
# NPP_im_MAX<-Vector_points[[nppBand_max]]
```

We need to define the years of the "warm up" phase. Remember that we will run one year at a time with different pools of data for each year.

```
# Define Year
year=seq(1/12,1,by=1/12)
```

We will need to set the RothC function to be ready to be used in the "Warm Up" process.

```
########### function set up
########### starts#################
Roth_C <- function(Cinputs, years, DPMptf,
    RPMptf, BIOptf, HUMptf, FallIOM, Temp,
    Precip, Evp, Cov, Cov1, Cov2, soil.thick,
    SOC, clay, DR, bare1, LU)  SOC,
    SOC, clay, DR, bare1, LU)  clay,
    SOC, clay, DR, bare1, LU)  DR,
    SOC, clay, DR, bare1, LU)  bare1,
    SOC, clay, DR, bare1, LU)  LU)
{

    # Paddy fields coefficent fPR = 0.4
    # if the target point is class = 13
    # , else fPR=1 From Shirato and
    # Yukozawa 2004

    fPR = (LU == 13) * 0.4 + (LU != 13) *
        1

    # Temperature effects per month
    fT = fT.RothC(Temp[, 2])

    # Moisture effects per month . If
    # using evapotranspiration pE=1
```

```
fw1func <- function(P, E, S.Thick = 30,
    pClay = 32.0213, pE = 1, bare) {

    M = P - E * pE
    Acc.TSMD = NULL
    for (i in 2:length(M)) {
        B = ifelse(bare[i] == FALSE,
            1, 1.8)
        Max.TSMD = -(20 + 1.3 * pClay -
            0.01 * (pClay^2)) * (S.Thick/23) *
            (1/B)
        Acc.TSMD[1] = ifelse(M[1] > 0,
            0, M[1])
        if (Acc.TSMD[i - 1] + M[i] <
            0) {
            Acc.TSMD[i] = Acc.TSMD[i -
              1] + M[i]
        } else (Acc.TSMD[i] = 0)
        if (Acc.TSMD[i] <= Max.TSMD) {
            Acc.TSMD[i] = Max.TSMD
        }
    }
    b = ifelse(Acc.TSMD > 0.444 * Max.TSMD,
        1, (0.2 + 0.8 * ((Max.TSMD -
            Acc.TSMD)/(Max.TSMD - 0.444 *
            Max.TSMD))))
    b <- clamp(b, lower = 0.2)
    return(data.frame(b))
}

fW_2 <- fw1func(P = (Precip[, 2]), E = (Evp[,
    2]), S.Thick = soil.thick, pClay = clay,
    pE = 1, bare = bare1)$b

# Vegetation Cover effects C1: No
# till Agriculture, C2:
# Conventional Agriculture, C3:
# Grasslands and Forests, C4
```

```
    # bareland and Urban

    fC <- Cov2[, 2]

    # Set the factors frame for Model
    # calculations

    xi.frame = data.frame(years, rep(fT *
        fW_2 * fC * fPR, length.out = length(years)))

    # RUN THE MODEL from SoilR Loads
    # the model if pass=TRUE also
    # invalid results are computed.
    # Model3_spin=RothCModel(t=years,C0=c(DPMptf,
    # RPMptf, BIOptf, HUMptf,
    # FallIOM),In=Cinputs,DR=DR,clay=clay,xi=xi.frame,
    # pass=TRUE) Calculates stocks for
    # each pool per month
    # Ct3_spin=getC(Model3_spin)

    # RUN THE MODEL from soilassesment

    Model3_spin = carbonTurnover(tt = years,
        C0 = c(DPMptf, RPMptf, BIOptf, HUMptf,
            FallIOM), In = Cinputs, Dr = DR,
        clay = clay, effcts = xi.frame, "euler")

    Ct3_spin = Model3_spin[, 2:6]

    # Get the final pools of the time
    # series

    poolSize3_spin = as.numeric(tail(Ct3_spin,
        1))

    return(poolSize3_spin)

}
```

```
############## funtion set up
############## ends##########
```

Then, we will apply the function for each target point 18/20 times according to the number of years of the "warm up" process. We will also create empty variables called "Cinputs","Cinputs_min", "Cinputs_max","NPP_M",

```
# Iterates over the area of interest and over 18 years
Cinputs<-c()
Cinputs_min<-c()
Cinputs_max<-c()
NPP_M_MIN<-c()
NPP_M_MAX<-c()
NPP_M<-c()
```

At this step, the iteration over the number of years of the warm up process is started. A parallel script is used that allows simultaneous use of multiple cores:

```
n_cores <- detectCores() - 1
n_cores

# Create a Cluster
parallelCluster <- makeCluster(n_cores, type = "SOCK",
                               methods = FALSE)
setDefaultCluster(parallelCluster)
registerDoParallel(parallelCluster)

#listOfRows <- split(Vector_variables, sample(1:4,
  # nrow(Vector_variables), replace=T))
WARM_UP<-Vector

blocks<-round(length(Vector_points)/n_cores)

listOfRows<- split(Vector_points,
                   (seq(nrow(Vector_points))-1) %/% blocks)
listOfCin<-  split(WARM_UP, (seq(nrow(WARM_UP))-1) %/% blocks)
listofSp<- split(Spin_up, (seq(nrow(Spin_up))-1) %/% blocks)
```

```
listOfRows<-listOfRows[1:n_cores]
listOfCin<-listOfCin[1:n_cores]
listofSp<-listofSp[1:n_cores]

# Before continuing check that the lists have
# the same number of elements than the cores.
x<-list()
results<-c()
system.time({
results <- foreach(j=1:length(listOfRows),.inorder = FALSE,
      .combine = rbind,.packages= c('raster','rgdal',
                    "SoilR","soilassessment")) %dopar% {

  iteration<-function(Vector_variables2,WARM_UP2,Spin_up2)
  {
    Cinputs<-c()
    Cinputs_min<-c()
    Cinputs_max<-c()
    NPP_M_MIN<-c()
    NPP_M_MAX<-c()
    NPP_M<-c()
    for (i in 1:nrow(Vector_variables2@data))
    {
      # Extract the variables 1

      gt<-firstClimLayer
      gp<-gt+clim_layers
      gevp<-gp+clim_layers

      for (w in 1:(dim(TEMP)[3]/12)) {

        print(c("year:",w))
        # Extract the variables

        Vect<-as.data.frame(Vector_variables2@data[i,])

        Temp<-as.data.frame(t(Vect[gt:(gt+11)]))
        Temp<-data.frame(Month=1:12, Temp=Temp[,1])
```

```
        Precip<-as.data.frame(t(Vect[gp:(gp+11)]))
        Precip<-data.frame(Month=1:12, Precip=Precip[,1])

        Evp<-as.data.frame(t(Vect[gevp:(gevp+11)]))
        Evp<-data.frame(Month=1:12, Evp=Evp[,1])

        Cov<-as.data.frame(t(Vect[4:15]))
        Cov1<-data.frame(Cov=Cov[,1])
        Cov2<-data.frame(Month=1:12, Cov=Cov[,1])

        DR_im<-as.data.frame(t(Vect[nDR_beg:nDR_end]))
        # DR one per year according to LU
        DR_im<-data.frame(DR_im=DR_im[,1])

        gt<-gt+12
        gp<-gp+12
        gevp<-gevp+12

        SOC_im<-Vector_variables2@data[[i,2]]

        clay_im<-Vector_variables2@data[[i,3]]

        LU_im<-Vector_variables2@data[[i,16]]

        NPP_im<-Vector_variables2@data[[i,nppBand]]

        NPP_im_MIN<-Vector_variables2@data[[i,nppBand_min]]

        NPP_im_MAX<-Vector_variables2@data[[i,nppBand_max]]
```

This line will avoid running the model over points with unreliable values:

```
#Avoid calculus over Na values

     if (any(is.na(Evp[,2])) | any(is.na(Temp[,2])) |
         any(is.na(SOC_im)) | any(is.na(clay_im)) |
         any(is.na(Spin_up@data[i,3]))  | any(is.na(NPP_im)) |
         any(is.na(Precip[,2]))  |  any(is.na(Cov2[,2]))  |
```

```
        any(is.na(Cov1[,1]))  | any(is.na(DR_im[,1]))    |
        (SOC_im<0) | (clay_im<0) |
        (Spin_up@data[i,3]<=0) ) {WARM_UP2[i,2]<-0}else{

         # Get the variables from the vector

         soil.thick=30 #Soil thickness (org. layer topsoil (cm))
         SOC<-SOC_im      #Soil organic carbon in t/ha
         clay<-clay_im        #Percent clay %

         DR<-DR_im[w,1]     # DPM/RPM (decomposable vs
                            #resistant plant material)
         bare1<-(Cov1>0.8) # If the surface is bare or vegetated
         NPP_81_00<-NPP_im
         NPP_81_00_MIN<-NPP_im_MIN
         NPP_81_00_MAX<-NPP_im_MAX
```

We will calculate the NPP MIAMI value for each point and each year , and adjust the carbon inputs with the NPP values. The first Cinput value corresponds to the Cinput of equilibrium calculated in the Spin Up phase (Spin_up[i,3]).

```
# PHASE 2 : WARM UP .  years (w)

# Cinputs
T <- mean(Temp[, 2])
P <- sum(Precip[, 2])
NPP_M[w] <- NPPmodel(P, T, "miami") * (1/100) *
    0.5
NPP_M[w] <- (LU_im == 2 | LU_im == 12 | LU_im ==
    13) * NPP_M[w] * 0.53 + (LU_im == 4) *
    NPP_M[w] * 0.88 + (LU_im == 3 | LU_im ==
    5 | LU_im == 6 | LU_im == 8) * NPP_M[w] *
    0.72

if (w == 1) {
    Cinputs[w] <- (Spin_up2@data[i, 3]/NPP_81_00) *
        NPP_M[w]
} else {
```

```
    Cinputs[w] <- (Cinputs[[w - 1]]/NPP_M[w -
        1]) * NPP_M[w]
}
```

Then we will repeat the same code but this time changing the environmental variables to match the maximum and minimum values.

```
# Cinputs MIN

Tmin <- mean(Temp[, 2] * 1.02)
Pmin <- sum(Precip[, 2] * 0.95)
NPP_M_MIN[w] <- NPPmodel(Pmin, Tmin, "miami") *
    (1/100) * 0.5
NPP_M_MIN[w] <- (LU_im == 2 | LU_im == 12 |
    LU_im == 13) * NPP_M_MIN[w] * 0.53 +
    (LU_im == 4) * NPP_M_MIN[w] * 0.88 +
    (LU_im == 3 | LU_im == 5 | LU_im == 6 |
        LU_im == 8) * NPP_M_MIN[w] * 0.72

if (w == 1) {
    Cinputs_min[w] <- (Spin_up2@data[i, 10]/NPP_81_00) *
        NPP_M_MIN[w]
} else {
    Cinputs_min[w] <- (Cinputs_min[[w - 1]]/NPP_M_MIN[w -
        1]) * NPP_M_MIN[w]
}

# Cinputs MAX

Tmax <- mean(Temp[, 2] * 0.98)
Pmax <- sum(Precip[, 2] * 1.05)
NPP_M_MAX[w] <- NPPmodel(Pmax, Tmax, "miami") *
    (1/100) * 0.5
NPP_M_MAX[w] <- (LU_im == 2 | LU_im == 12 |
    LU_im == 13) * NPP_M_MAX[w] * 0.53 +
    (LU_im == 4) * NPP_M_MAX[w] * 0.88 +
    (LU_im == 3 | LU_im == 5 | LU_im == 6 |
        LU_im == 8) * NPP_M_MAX[w] * 0.72
```

```
if (w == 1) {
    Cinputs_max[w] <- (Spin_up2@data[i, 11]/NPP_81_00) *
        NPP_M_MAX[w]
} else {
    Cinputs_max[w] <- (Cinputs_max[[w - 1]]/NPP_M_MAX[w -
        1]) * NPP_M_MAX[w]
}
```

We will then run the RothC function for each point and each year. The first year we will use the equilibrium Cinputs, and the carbon pools obtained from the Spin Up phase. Then we will use the yearly adjusted Cinputs (using NPP) and the pools calculated from the previous iteration.

```
# Run the model for 2001-2018

          if (w==1) {
            f_wp<-Roth_C(Cinputs=Cinputs[1],years=years,
                         DPMptf=Spin_up2@data[i,5],
                         RPMptf=Spin_up2@data[i,6],
                         BIOptf=Spin_up2@data[i,7],
                         HUMptf=Spin_up2@data[i,8],
                         FallIOM=Spin_up2@data[i,9],
                         Temp=Temp,Precip=Precip,Evp=Evp,
                         Cov=Cov,Cov1=Cov1,Cov2=Cov2,
                         soil.thick=soil.thick,SOC=SOC,
                         clay=clay,DR=DR,bare1=bare1,
                         LU=LU_im)
          } else {
            f_wp<-Roth_C(Cinputs=Cinputs[w],years=years,
                         DPMptf=f_wp[1], RPMptf=f_wp[2],
                         BIOptf=f_wp[3], HUMptf=f_wp[4],
                         FallIOM=f_wp[5],Temp=Temp,
                         Precip=Precip,Evp=Evp,Cov=Cov,
                         Cov1=Cov1,Cov2=Cov2,
                         soil.thick=soil.thick,SOC=SOC,
                         clay=clay,DR=DR,bare1=bare1,
                         LU=LU_im)
          }
```

```
f_wp_t<-f_wp[1]+f_wp[2]+f_wp[3]+f_wp[4]+f_wp[5]

# Run the model for minimum values

if (w==1) {
  f_wp_min<-Roth_C(Cinputs=Cinputs_min[1],
    years=years,DPMptf=Spin_up2@data[i,13],
    RPMptf=Spin_up2@data[i,14],
    BIOptf=Spin_up2@data[i,15],
    HUMptf=Spin_up2@data[i,16],
    FallIOM=Spin_up2@data[i,17],Temp=Temp*1.02,
    Precip=Precip*0.95,Evp=Evp,Cov=Cov,Cov1=Cov1,
    Cov2=Cov2,soil.thick=soil.thick,SOC=SOC*0.8,
    clay=clay*0.9,DR=DR,bare1=bare1,LU=LU_im)
} else {
  f_wp_min<-Roth_C(Cinputs=Cinputs_min[w],
    years=years,DPMptf=f_wp_min[1],
    RPMptf=f_wp_min[2], BIOptf=f_wp_min[3],
    HUMptf=f_wp_min[4], FallIOM=f_wp_min[5],
    Temp=Temp*1.02,Precip=Precip*0.95,Evp=Evp,
    Cov=Cov,Cov1=Cov1,Cov2=Cov2,
    soil.thick=soil.thick,SOC=SOC*0.8,
    clay=clay*0.9,DR=DR,bare1=bare1,LU=LU_im)
}

f_wp_t_min<-f_wp_min[1]+f_wp_min[2]+f_wp_min[3]+
  f_wp_min[4]+f_wp_min[5]

# Run the model for maximum values

if (w==1) {
  f_wp_max<-Roth_C(Cinputs=Cinputs_max[1],
    years=years,DPMptf=Spin_up2@data[i,19],
    RPMptf=Spin_up2@data[i,20],
    BIOptf=Spin_up2@data[i,21],
    HUMptf=Spin_up2@data[i,22],
    FallIOM=Spin_up2@data[i,23],Temp=Temp*0.98,
    Precip=Precip*1.05,Evp=Evp,Cov=Cov,Cov1=Cov1,
```

```
            Cov2=Cov2,soil.thick=soil.thick,SOC=SOC*1.2,
            clay=clay*1.1,DR=DR,bare1=bare1,LU=LU_im)
        } else {
          f_wp_max<-Roth_C(Cinputs=Cinputs_max[w],
            years=years,DPMptf=f_wp_max[1],
            RPMptf=f_wp_max[2], BIOptf=f_wp_max[3],
            HUMptf=f_wp_max[4], FallIOM=f_wp_max[5],
            Temp=Temp*0.98,Precip=Precip*1.05,Evp=Evp,
            Cov=Cov,Cov1=Cov1,Cov2=Cov2,soil.thick=soil.thick,
            SOC=SOC*1.2,clay=clay*1.1,DR=DR,
            bare1=bare1,LU=LU_im)
        }

        f_wp_t_max<-f_wp_max[1]+f_wp_max[2]+f_wp_max[3]+
          f_wp_max[4]+f_wp_max[5]

        #print(w)
        #print(c(i,SOC,Spin_up[i,3],NPP_81_00,
        # Cinputs[w],f_wp_t))
        ##print(c(NPP_M[w],Cinputs[w]))
      }
    }
```

We will save the results from the iteration of the last year to the empty vector. We will also calculate an average of all the Cinputs used in the warm up phase and save it. We will need this "CinputFORWARD" variable in the next phase (Forward) and script (script number 15).

```
if (is.na(mean(Cinputs))){CinputFORWARD<-NA} else {

        CinputFORWARD<-mean(Cinputs)

        CinputFORWARD_min<-mean(Cinputs_min)

        CinputFORWARD_max<-mean(Cinputs_max)

        WARM_UP2[i,2]<-SOC
        WARM_UP2[i,3]<-Cinputs[18]
```

```
        WARM_UP2[i,4]<-f_wp_t
        WARM_UP2[i,5]<-f_wp[1]
        WARM_UP2[i,6]<-f_wp[2]
        WARM_UP2[i,7]<-f_wp[3]
        WARM_UP2[i,8]<-f_wp[4]
        WARM_UP2[i,9]<-f_wp[5]
        WARM_UP2[i,10]<-CinputFORWARD
        WARM_UP2[i,11]<-f_wp_t_min
        WARM_UP2[i,12]<-f_wp_min[1]
        WARM_UP2[i,13]<-f_wp_min[2]
        WARM_UP2[i,14]<-f_wp_min[3]
        WARM_UP2[i,15]<-f_wp_min[4]
        WARM_UP2[i,16]<-f_wp_min[5]
        WARM_UP2[i,17]<-f_wp_t_max
        WARM_UP2[i,18]<-f_wp_max[1]
        WARM_UP2[i,19]<-f_wp_max[2]
        WARM_UP2[i,20]<-f_wp_max[3]
        WARM_UP2[i,21]<-f_wp_max[4]
        WARM_UP2[i,22]<-f_wp_max[5]
        WARM_UP2[i,23]<-CinputFORWARD_min
        WARM_UP2[i,24]<-CinputFORWARD_max

        Cinputs<-c()
        Cinputs_min<-c()
        Cinputs_max<-c()


      }


    }
    return(WARM_UP2)}

  x<- iteration(Vector_variables2=listOfRows[[j]],
        WARM_UP2=listOfCin[[j]],Spin_up2=listofSp[[j]])
  x
}
```

```
})
# Stop the clusters
stopCluster(parallelCluster)


WARM_UP<-results

################for loop ends#############
```

We will then run the last code block to change the names to the fields of the vector's table.

```
colnames(WARM_UP@data)[2]="SOC_FAO"
colnames(WARM_UP@data)[3]="Cin_t0"
colnames(WARM_UP@data)[4]="SOC_t0"
colnames(WARM_UP@data)[5]="DPM_w_up"
colnames(WARM_UP@data)[6]="RPM_w_up"
colnames(WARM_UP@data)[7]="BIO_w_up"
colnames(WARM_UP@data)[8]="HUM_w_up"
colnames(WARM_UP@data)[9]="IOM_w_up"
colnames(WARM_UP@data)[10]="Cin_mean"
colnames(WARM_UP@data)[11]="SOC_t0min"
colnames(WARM_UP@data)[12]="DPM_w_min"
colnames(WARM_UP@data)[13]="RPM_w_min"
colnames(WARM_UP@data)[14]="BIO_w_min"
colnames(WARM_UP@data)[15]="HUM_w_min"
colnames(WARM_UP@data)[16]="IOM_w_min"
colnames(WARM_UP@data)[17]="SOC_t0max"
colnames(WARM_UP@data)[18]="DPM_w_max"
colnames(WARM_UP@data)[19]="RPM_w_max"
colnames(WARM_UP@data)[20]="BIO_w_max"
colnames(WARM_UP@data)[21]="HUM_w_max"
colnames(WARM_UP@data)[22]="IOM_w_max"
colnames(WARM_UP@data)[23]="Cin_min"
colnames(WARM_UP@data)[24]="Cin_max"
```

Finally, we will have to save the output vector and the name of that vector.

```
# SAVE the Points (shapefile)
setwd("C:/TRAINING_MATERIALS/OUTPUTS/2_WARM_UP")
writeOGR(WARM_UP,".", "WARM_UP_Country_AOI",
         driver="ESRI Shapefile",overwrite=TRUE)
```

10.4 Forward phase

10.4.1 Script Number 15."ROTH_C_forward_UNC_v4_ parallel.R"

Script number 15 implements the third modeling phase ("forward" phase). We will need to load the stack of layers generated in script number 12 and the target points. We will also need to load the output vector of the phase 2 ("warm up") as an input. This script will run the Roth C model for 20 years, projecting SOC stocks for the 2020-2040 period under different management scenarios ("BAU" scenario and the three SSM scenarios: low, medium and high input carbon). C inputs will vary according to the SSM scenarios. Standard default values of 5-10-20 percent increase in C inputs are defined for the three SSM scenarios (low, medium, high, respectively). Users can modify these inputs based on local expertise and available information, and generate alternative maps using this data. The final outputs will be the final SOC stocks after 20 years for the different scenarios. This information will be saved to a shapefile.

RothC froward modeling.

Input	Output
Point vector with the locations to run the model (empty vector, should come from the SOCmap FAO, one point per pixel). (from QGIS procedure 1) WARM_UP.shp (script number 14)	FORWARD_BAU_3E_20YEARS_[code_country].shp (contains the output of model for BAU, and three future scenarios based on Carbon input improvement)
Stack layer (from script number 12): Stack_Set_FORWARD_[Country_code].tif	

The 'Forward' modeling phase requires (as in the previous phases) the target points (generated from the Qgis procedure number 1), the stack of layers (from script number 12), and the output vector from the previous phase (warm up).

We will need to load the R packages, the target points, the stack for this phase (Stack_Set_forward_[country_code].tif), and vector from the 'Warm up' phase (WARM_UP.shp).

```
rm(list = ls())

library(SoilR)
library(raster)
library(rgdal)
library(soilassessment)

WD_OUT <- ("C:/TRAINING_MATERIALS/OUTPUTS/3_FOWARD")

working_dir <- setwd("C:/TRAINING_MATERIALS")

# OPEN THE VECTOR OF POINTS

Vector <- readOGR("INPUTS/TARGET_POINTS/target_points_sub.shp")

# OPEN THE RESULT VECTOR FROM THE WARM
# UP PROCESS

WARM_UP <- readOGR("OUTPUTS/2_WARM_UP/WARM_UP_Country_AOI.shp")

# OPEN THE STACK WITH THE VARIABLES FOR
# THE FORWARD PROCESS

Stack_Set_1 <- stack("INPUTS/STACK/Stack_Set_FORWARD.tif")
```

We can set the SSM carbon input increment of each land use class.

```
# Set the increase in Carbon input for
# each land use and each scenario

# Crops and Crop trees
Low_Crops <- 1.05
Med_Crops <- 1.1
High_Crops <- 1.2
```

```
# Shrublands, Grasslands , Herbaceous
# vegetation flooded & Sparse
# Vegetation
Low_Grass <- 1.05
Med_Grass <- 1.1
High_Grass <- 1.2

# Forest
Low_Forest <- 1.05
Med_Forest <- 1.1
High_Forest <- 1.2

# Paddy Fields
Low_PaddyFields <- 1.05
Med_PaddyFields <- 1.1
High_PaddyFields <- 1.2
```

Next, we will extract the variables contained in the stack to the target points.

```
# extract variables to points
Variables<-extract(Stack_Set_1,Vector,sp=TRUE)
```

Then we will create a variable to save the results from the simulations.

```
# Creates an empty vector
FORWARD<-Vector
```

Next, we will define the number of years for the modeling runs.

```
# Define the years to run the model
years=seq(1/12,20,by=1/12)
```

Then, we will set the RothC function as we did for the previous modeling phases.

```
############## function set up
############## starts###############
```

```
Roth_C <- function(Cinputs, years, DPMptf,
    RPMptf, BIOptf, HUMptf, FallIOM, Temp,
    Precip, Evp, Cov, Cov1, Cov2, soil.thick,
    SOC, clay, DR, bare1, LU) {
    # Paddy Fields coefficent fPR = 0.4
    # if the target point is class = 13
    # , else fPR=1 From Shirato and
    # Yukozawa 2004

    fPR = (LU == 13) * 0.4 + (LU != 13) *
        1

    # Temperature effects per month
    fT = fT.RothC(Temp[, 2])

    # Moisture effects per month .

    fw1func <- function(P, E, S.Thick = 30,
        pClay = 32.0213, pE = 1, bare) {

        M = P - E * pE
        Acc.TSMD = NULL
        for (i in 2:length(M)) {
            B = ifelse(bare[i] == FALSE,
                1, 1.8)
            Max.TSMD = -(20 + 1.3 * pClay -
                0.01 * (pClay^2)) * (S.Thick/23) *
                (1/B)
            Acc.TSMD[1] = ifelse(M[1] > 0,
                0, M[1])
            if (Acc.TSMD[i - 1] + M[i] <
                0) {
                Acc.TSMD[i] = Acc.TSMD[i -
                  1] + M[i]
            } else (Acc.TSMD[i] = 0)
            if (Acc.TSMD[i] <= Max.TSMD) {
                Acc.TSMD[i] = Max.TSMD
            }
```

```
        }
        b = ifelse(Acc.TSMD > 0.444 * Max.TSMD,
            1, (0.2 + 0.8 * ((Max.TSMD -
                Acc.TSMD)/(Max.TSMD - 0.444 *
                Max.TSMD))))
        b <- clamp(b, lower = 0.2)
        return(data.frame(b))
    }

    fW_2 <- fw1func(P = (Precip[, 2]), E = (Evp[,
        2]), S.Thick = soil.thick, pClay = clay,
        pE = 1, bare = bare1)$b

    # Vegetation Cover effects

    fC <- Cov2[, 2]

    # Set the factors frame for Model
    # calculations

    xi.frame = data.frame(years, rep(fT *
        fW_2 * fC * fPR, length.out = length(years)))

    # RUN THE MODEL from SoilR Loads
    # the model
    # Model3_spin=RothCModel(t=years,C0=c(DPMptf[[1]],
    # RPMptf[[1]], BIOptf[[1]],
    # HUMptf[[1]],
    # FallIOM[[1]]),In=Cinputs,DR=DR,clay=clay,xi=xi.frame,
    # pass=TRUE)
    # Ct3_spin=getC(Model3_spin)

    # RUN THE MODEL from soilassesment

    Model3_spin = carbonTurnover(tt = years,
        C0 = c(DPMptf[[1]], RPMptf[[1]],
            BIOptf[[1]], HUMptf[[1]], FallIOM[[1]]),
        In = Cinputs, Dr = DR, clay = clay,
```

```
        effcts = xi.frame, "euler")

    Ct3_spin = Model3_spin[, 2:6]

    # Get the final pools of the time
    # series

    poolSize3_spin = as.numeric(tail(Ct3_spin,
        1))

    return(poolSize3_spin)
}
################ function set up
################ ends##############
```

Then we can iterate over the vector variable, running the model for each target point using parallel scripts.

```
library(foreach)
library(doParallel)

n_cores <- detectCores() - 1
n_cores

# we create a cluster
parallelCluster <- makeCluster(n_cores, type = "SOCK",
                               methods = FALSE)

setDefaultCluster(parallelCluster)
registerDoParallel(parallelCluster)

blocks<-round(length(Variables)/n_cores)

listOfRows<- split(Variables, (seq(nrow(Variables))-1) %/% blocks)
listOfCin<-  split(FORWARD, (seq(nrow(FORWARD))-1) %/% blocks)
listOfWup<-  split(WARM_UP, (seq(nrow(WARM_UP))-1) %/% blocks)

listOfRows<-listOfRows[1:n_cores]
```

```
listOfCin<-listOfCin[1:n_cores]
listOfWup<-listOfWup[1:n_cores]

#Check that the  lists have the same number of blocks of
#the cores that you are going to use.
# If not reduce the number of cores.

x<-list()
results<-c()

# Iterates over the area of interest
###################for loop starts################
results <- foreach(j=1:length(listOfRows),.inorder = FALSE,
              .combine = rbind,.packages= c('raster',
              'rgdal',"SoilR","soilassessment")) %dopar% {

  iteration<-function(Vector_variables2,FORWARD2,WARM_UP2)
  {


for (i in 1:dim(Vector_variables2@data)){

# Extract the variables

Vect<-as.data.frame(Vector_variables2@data[i,])

Temp<-as.data.frame(t(Vect[4:15]))
Temp<-data.frame(Month=1:12, Temp=Temp[,1])

Precip<-as.data.frame(t(Vect[16:27]))
Precip<-data.frame(Month=1:12, Precip=Precip[,1])

Evp<-as.data.frame(t(Vect[28:39]))
Evp<-data.frame(Month=1:12, Evp=Evp[,1])

Cov<-as.data.frame(t(Vect[42:53]))
Cov1<-data.frame(Cov=Cov[,1])
Cov2<-data.frame(Month=1:12, Cov=Cov[,1])
```

```
SOC_im<-WARM_UP2@data[[i,4]]

clay_im<-Vector_variables2@data[[i,3]]

LU_im<-Vector_variables2@data[[i,41]]

Cinputs_im<-WARM_UP2@data[[i,10]]

DR_im<-Vector_variables2@data[[i,40]]

#Avoid calculus over Na values

if (any(is.na(Evp[,2])) | any(is.na(Temp[,2])) |
    any(is.na(SOC_im)) | any(is.na(clay_im)) |
    any(is.na(Precip[,2]))  |  any(is.na(Cov2[,2])) |
    any(is.na(Cov1[,1])) | any(is.na(Cinputs_im)) |
    any(is.na(DR_im)) | (Cinputs_im<0) |  (SOC_im<0) |
    (clay_im<0) ) {FORWARD2[i,2]<-0}else{

  # Set the variables from the images

soil.thick=30  #Soil thickness (organic layer topsoil), in cm
SOC<-SOC_im      #Soil organic carbon in Mg/ha
clay<-clay_im      #Percent clay %
Cinputs<-Cinputs_im    #Annual C inputs to soil in Mg/ha/yr

DR<-DR_im   # DPM/RPM (decomposable vs resistant plant material)
bare1<-(Cov1>0.8)  # If the surface is bare or vegetated
LU<-LU_im
```

Now we can run the model for the business as usual (BAU) scenario. The "f_bau_t" variable will save the result of the model in each iteration.

```
# Final calculation of SOC  20 years in the future
# (Business as usual)

f_bau<-Roth_C(Cinputs=Cinputs,years=years,
              DPMptf=WARM_UP2@data[[i,5]],
```

```
                RPMptf=WARM_UP2@data[[i,6]],
                BIOptf=WARM_UP2@data[[i,7]],
                HUMptf=WARM_UP2@data[[i,8]],
                FallIOM=WARM_UP2@data[[i,9]],
                Temp=Temp,Precip=Precip,Evp=Evp,
                Cov=Cov,Cov1=Cov1,Cov2=Cov2,
                soil.thick=soil.thick,SOC=SOC,
                clay=clay,DR=DR,bare1=bare1,LU=LU)
f_bau_t<-f_bau[1]+f_bau[2]+f_bau[3]+f_bau[4]+f_bau[5]

#Unc BAU minimum
Cinputs_min<-WARM_UP2@data[i,23]
Cinputs_max<-WARM_UP2@data[i,24]
SOC_t0_min<-WARM_UP2@data[i,11]
SOC_t0_max<-WARM_UP2@data[i,17]

f_bau_min<-Roth_C(Cinputs=Cinputs_min,years=years,
                  DPMptf=WARM_UP2@data[[i,12]],
                  RPMptf=WARM_UP2@data[[i,13]],
                  BIOptf=WARM_UP2@data[[i,14]],
                  HUMptf=WARM_UP2@data[[i,15]],
                  FallIOM=WARM_UP2@data[[i,16]],
                  Temp=Temp*1.02,Precip=Precip*0.95,
                  Evp=Evp,Cov=Cov,Cov1=Cov1,Cov2=Cov2,
                  soil.thick=soil.thick,SOC=SOC*0.8,
                  clay=clay*0.9,DR=DR,bare1=bare1,LU=LU)
f_bau_t_min<-f_bau_min[1]+f_bau_min[2]+f_bau_min[3]+
  f_bau_min[4]+f_bau_min[5]

#Unc BAU maximum

f_bau_max<-Roth_C(Cinputs=Cinputs_max,years=years,
                  DPMptf=WARM_UP2@data[[i,18]],
                  RPMptf=WARM_UP2@data[[i,19]],
                  BIOptf=WARM_UP2@data[[i,20]],
                  HUMptf=WARM_UP2@data[[i,21]],
                  FallIOM=WARM_UP2@data[[i,22]],
                  Temp=Temp*0.98,Precip=Precip*1.05,
```

```
                  Evp=Evp,Cov=Cov,Cov1=Cov1,Cov2=Cov2,
                  soil.thick=soil.thick,SOC=SOC*1.2,
                  clay=clay*1.1,DR=DR,bare1=bare1,LU=LU)
f_bau_t_max<-f_bau_max[1]+f_bau_max[2]+f_bau_max[3]+
  f_bau_max[4]+f_bau_max[5]

# Crops and Tree crops
if (LU==2 | LU==12){
f_low<-Roth_C(Cinputs=(Cinputs*Low_Crops),years=years,
              DPMptf=WARM_UP2@data[[i,5]],
              RPMptf=WARM_UP2@data[[i,6]],
              BIOptf=WARM_UP2@data[[i,7]],
              HUMptf=WARM_UP2@data[[i,8]],
              FallIOM=WARM_UP2@data[[i,9]],
              Temp=Temp,Precip=Precip,Evp=Evp,Cov=Cov,
              Cov1=Cov1,Cov2=Cov2,soil.thick=soil.thick,
              SOC=SOC,clay=clay,DR=DR,bare1=bare1,LU=LU)
f_low_t<-f_low[1]+f_low[2]+f_low[3]+f_low[4]+f_low[5]

f_med<-Roth_C(Cinputs=(Cinputs*Med_Crops),years=years,
              DPMptf=WARM_UP2@data[[i,5]],
              RPMptf=WARM_UP2@data[[i,6]],
              BIOptf=WARM_UP2@data[[i,7]],
              HUMptf=WARM_UP2@data[[i,8]],
              FallIOM=WARM_UP2@data[[i,9]],
              Temp=Temp,Precip=Precip,Evp=Evp,Cov=Cov,
              Cov1=Cov1,Cov2=Cov2,soil.thick=soil.thick,
              SOC=SOC,clay=clay,DR=DR,bare1=bare1,LU=LU)
f_med_t<-f_med[1]+f_med[2]+f_med[3]+f_med[4]+f_med[5]

f_high<-Roth_C(Cinputs=(Cinputs*High_Crops),years=years,
               DPMptf=WARM_UP2@data[[i,5]],
               RPMptf=WARM_UP2@data[[i,6]],
               BIOptf=WARM_UP2@data[[i,7]],
               HUMptf=WARM_UP2@data[[i,8]],
               FallIOM=WARM_UP2@data[[i,9]],
               Temp=Temp,Precip=Precip,Evp=Evp,Cov=Cov,
               Cov1=Cov1,Cov2=Cov2,soil.thick=soil.thick,
```

```
              SOC=SOC,clay=clay,DR=DR,bare1=bare1,LU=LU)
f_high_t<-f_high[1]+f_high[2]+f_high[3]+f_high[4]+f_high[5]

# SSM croplands unc min

f_med_min<-Roth_C(Cinputs=(Cinputs_min*(Med_Crops-0.15)),
                  years=years,DPMptf=WARM_UP2@data[[i,12]],
                  RPMptf=WARM_UP2@data[[i,13]],
                  BIOptf=WARM_UP2@data[[i,14]],
                  HUMptf=WARM_UP2@data[[i,15]],
                  FallIOM=WARM_UP2@data[[i,16]],
                  Temp=Temp*1.02,Precip=Precip*0.95,Evp=Evp,
                  Cov=Cov,Cov1=Cov1,Cov2=Cov2,
                  soil.thick=soil.thick,SOC=SOC*0.8,
                  clay=clay*0.9,DR=DR,bare1=bare1,LU=LU)
f_med_t_min<-f_med_min[1]+f_med_min[2]+f_med_min[3]+
  f_med_min[4]+f_med_min[5]

# SSM croplands unc max

f_med_max<-Roth_C(Cinputs=(Cinputs_max*(Med_Crops+0.15)),
                  years=years,DPMptf=WARM_UP2@data[[i,18]],
                  RPMptf=WARM_UP2@data[[i,19]],
                  BIOptf=WARM_UP2@data[[i,20]],
                  HUMptf=WARM_UP2@data[[i,21]],
                  FallIOM=WARM_UP2@data[[i,22]],
                  Temp=Temp*0.98,Precip=Precip*1.05,Evp=Evp,
                  Cov=Cov,Cov1=Cov1,Cov2=Cov2,
                  soil.thick=soil.thick,
                  SOC=SOC*1.2,clay=clay*1.1,DR=DR,
                  bare1=bare1,LU=LU)
f_med_t_max<-f_med_max[1]+f_med_max[2]+f_med_max[3]+
  f_med_max[4]+f_med_max[5]

}
#Shrublands, grasslands, and sparse vegetation
else if (LU==3 | LU==5 | LU==6 | LU==8) {
f_low<-Roth_C(Cinputs=(Cinputs*Low_Grass),years=years,
```

```
                DPMptf=WARM_UP2@data[[i,5]],
                RPMptf=WARM_UP2@data[[i,6]],
                BIOptf=WARM_UP2@data[[i,7]],
                HUMptf=WARM_UP2@data[[i,8]],
                FallIOM=WARM_UP2@data[[i,9]],Temp=Temp,
                Precip=Precip,Evp=Evp,Cov=Cov,Cov1=Cov1,
                Cov2=Cov2,soil.thick=soil.thick,SOC=SOC,
                clay=clay,DR=DR,bare1=bare1,LU=LU)
f_low_t<-f_low[1]+f_low[2]+f_low[3]+f_low[4]+f_low[5]

f_med<-Roth_C(Cinputs=(Cinputs*Med_Grass),years=years,
                DPMptf=WARM_UP2@data[[i,5]],
                RPMptf=WARM_UP2@data[[i,6]],
                BIOptf=WARM_UP2@data[[i,7]],
                HUMptf=WARM_UP2@data[[i,8]],
                FallIOM=WARM_UP2@data[[i,9]],Temp=Temp,
                Precip=Precip,Evp=Evp,Cov=Cov,Cov1=Cov1,
                Cov2=Cov2,soil.thick=soil.thick,SOC=SOC,
                clay=clay,DR=DR,bare1=bare1,LU=LU)
f_med_t<-f_med[1]+f_med[2]+f_med[3]+f_med[4]+f_med[5]

f_high<-Roth_C(Cinputs=(Cinputs*High_Grass),years=years,
                 DPMptf=WARM_UP2@data[[i,5]],
                 RPMptf=WARM_UP2@data[[i,6]],
                 BIOptf=WARM_UP2@data[[i,7]],
                 HUMptf=WARM_UP2@data[[i,8]],
                 FallIOM=WARM_UP2@data[[i,9]],
                 Temp=Temp,Precip=Precip,Evp=Evp,Cov=Cov,
                 Cov1=Cov1,Cov2=Cov2,soil.thick=soil.thick,
                 SOC=SOC,clay=clay,DR=DR,bare1=bare1,LU=LU)
f_high_t<-f_high[1]+f_high[2]+f_high[3]+f_high[4]+f_high[5]

#SSM Shrublands unc min

f_med_min<-Roth_C(Cinputs=(Cinputs_min*(Med_Grass-0.15)),
                   years=years,DPMptf=WARM_UP2@data[[i,12]],
                   RPMptf=WARM_UP2@data[[i,13]],
                   BIOptf=WARM_UP2@data[[i,14]],
```

```
                  HUMptf=WARM_UP2@data[[i,15]],
                  FallIOM=WARM_UP2@data[[i,16]],
                  Temp=Temp*1.02,Precip=Precip*0.95,Evp=Evp,
                  Cov=Cov,Cov1=Cov1,Cov2=Cov2,
                  soil.thick=soil.thick,
                  SOC=SOC*0.8,clay=clay*0.9,DR=DR,
                  bare1=bare1,LU=LU)
f_med_t_min<-f_med_min[1]+f_med_min[2]+f_med_min[3]+
  f_med_min[4]+f_med_min[5]

#SSM Shrublands unc max

f_med_max<-Roth_C(Cinputs=(Cinputs_max*(Med_Grass+0.15)),
                  years=years,DPMptf=WARM_UP2@data[[i,18]],
                  RPMptf=WARM_UP2@data[[i,19]],
                  BIOptf=WARM_UP2@data[[i,20]],
                  HUMptf=WARM_UP2@data[[i,21]],
                  FallIOM=WARM_UP2@data[[i,22]],
                  Temp=Temp*0.98,Precip=Precip*1.05,
                  Evp=Evp,Cov=Cov,Cov1=Cov1,Cov2=Cov2,
                  soil.thick=soil.thick,SOC=SOC*1.2,
                  clay=clay*1.1,DR=DR,bare1=bare1,LU=LU)
f_med_t_max<-f_med_max[1]+f_med_max[2]+f_med_max[3]+
  f_med_max[4]+f_med_max[5]

}
# Paddy Fields
else if (LU==13) {
f_low<-Roth_C(Cinputs=(Cinputs*Low_PaddyFields),years=years,
              DPMptf=WARM_UP2@data[[i,5]],
              RPMptf=WARM_UP2@data[[i,6]],
              BIOptf=WARM_UP2@data[[i,7]],
              HUMptf=WARM_UP2@data[[i,8]],
              FallIOM=WARM_UP2@data[[i,9]],Temp=Temp,
              Precip=Precip,Evp=Evp,Cov=Cov,Cov1=Cov1,
              Cov2=Cov2,soil.thick=soil.thick,SOC=SOC,
              clay=clay,DR=DR,bare1=bare1,LU=LU)
f_low_t<-f_low[1]+f_low[2]+f_low[3]+f_low[4]+f_low[5]
```

```
f_med<-Roth_C(Cinputs=(Cinputs*Med_PaddyFields),years=years,
              DPMptf=WARM_UP2@data[[i,5]],
              RPMptf=WARM_UP2@data[[i,6]],
              BIOptf=WARM_UP2@data[[i,7]],
              HUMptf=WARM_UP2@data[[i,8]],
              FallIOM=WARM_UP2@data[[i,9]],Temp=Temp,
              Precip=Precip,Evp=Evp,Cov=Cov,Cov1=Cov1,
              Cov2=Cov2,soil.thick=soil.thick,SOC=SOC,
              clay=clay,DR=DR,bare1=bare1,LU=LU)
f_med_t<-f_med[1]+f_med[2]+f_med[3]+f_med[4]+f_med[5]

f_high<-Roth_C(Cinputs=(Cinputs*High_PaddyFields),years=years,
               DPMptf=WARM_UP2@data[[i,5]],
               RPMptf=WARM_UP2@data[[i,6]],
               BIOptf=WARM_UP2@data[[i,7]],
               HUMptf=WARM_UP2@data[[i,8]],
               FallIOM=WARM_UP2@data[[i,9]],Temp=Temp,
               Precip=Precip,Evp=Evp,Cov=Cov,Cov1=Cov1,
               Cov2=Cov2,soil.thick=soil.thick,SOC=SOC,
               clay=clay,DR=DR,bare1=bare1,LU=LU)
f_high_t<-f_high[1]+f_high[2]+f_high[3]+f_high[4]+f_high[5]

#SSM Paddy Fields unc min

f_med_min<-Roth_C(Cinputs=(Cinputs_min*(Med_PaddyFields-0.15)),
                  years=years,DPMptf=WARM_UP2@data[[i,12]],
                  RPMptf=WARM_UP2@data[[i,13]],
                  BIOptf=WARM_UP2@data[[i,14]],
                  HUMptf=WARM_UP2@data[[i,15]],
                  FallIOM=WARM_UP2@data[[i,16]],
                  Temp=Temp*1.02,Precip=Precip*0.95,Evp=Evp,
                  Cov=Cov,Cov1=Cov1,Cov2=Cov2,
                  soil.thick=soil.thick,
                  SOC=SOC*0.8,clay=clay*0.9,DR=DR,
                  bare1=bare1,LU=LU)
f_med_t_min<-f_med_min[1]+f_med_min[2]+f_med_min[3]+
  f_med_min[4]+f_med_min[5]
```

```
#SSM Paddy Fields unc max

f_med_max<-Roth_C(Cinputs=(Cinputs_max*(Med_PaddyFields+0.15)),
                  years=years,DPMptf=WARM_UP2@data[[i,18]],
                  RPMptf=WARM_UP2@data[[i,19]],
                  BIOptf=WARM_UP2@data[[i,20]],
                  HUMptf=WARM_UP2@data[[i,21]],
                  FallIOM=WARM_UP2@data[[i,22]],Temp=Temp*0.98,
                  Precip=Precip*1.05,Evp=Evp,Cov=Cov,Cov1=Cov1,
                  Cov2=Cov2,soil.thick=soil.thick,SOC=SOC*1.2,
                  clay=clay*1.1,DR=DR,bare1=bare1,LU=LU)
f_med_t_max<-f_med_max[1]+f_med_max[2]+f_med_max[3]+
  f_med_max[4]+f_med_max[5]

}

else{
f_bau_t<-0
f_low_t<-0
f_med_t<-0
f_high_t<-0
f_bau_t_min<-0
f_bau_t_max<-0
f_med_t_min<-0
f_med_t_max<-0
SOC_t0_min<-0
SOC_t0_max<-0

}

FORWARD2[i,2]<-SOC
FORWARD2[i,3]<-f_bau_t
FORWARD2[i,4]<-f_bau[1]
FORWARD2[i,5]<-f_bau[2]
FORWARD2[i,6]<-f_bau[3]
FORWARD2[i,7]<-f_bau[4]
FORWARD2[i,8]<-f_bau[5]
FORWARD2[i,9]<-LU
```

```
FORWARD2[i,10]<-f_low_t
FORWARD2[i,11]<-f_med_t
FORWARD2[i,12]<-f_high_t
FORWARD2[i,13]<-f_bau_t_min
FORWARD2[i,14]<-f_bau_t_max
FORWARD2[i,15]<-f_med_t_min
FORWARD2[i,16]<-f_med_t_max
FORWARD2[i,17]<-SOC_t0_min
FORWARD2[i,18]<-SOC_t0_max



#print(c(i,SOC,f_bau_t,f_low_t,f_med_t,
# f_high_t,f_bau_t_min,f_bau_t_max))

}
}
    return(FORWARD2)}

  x<- iteration(Vector_variables2=listOfRows[[j]],
      FORWARD2=listOfCin[[j]],WARM_UP2=listOfWup[[j]])
  x
}


stopCluster(parallelCluster)


FORWARD<-results
#############for loop ends###############
```

Next , we will change the names of the fields in the output vector layer.

```
colnames(FORWARD@data)[2]="SOC_t0"
colnames(FORWARD@data)[3]="SOC_BAU_20"
colnames(FORWARD@data)[4]="DPM_BAU_20"
colnames(FORWARD@data)[5]="RPM_BAU_20"
colnames(FORWARD@data)[6]="BIO_BAU_20"
```

```
colnames(FORWARD@data)[7]="HUM_BAU_20"
colnames(FORWARD@data)[8]="IOM_BAU_20"
colnames(FORWARD@data)[9]="LandUse"
colnames(FORWARD@data)[10]="Low_Scenario"
colnames(FORWARD@data)[11]="Med_Scenario"
colnames(FORWARD@data)[12]="High_Scenario"
colnames(FORWARD@data)[13]="SOC_BAU_20_min"
colnames(FORWARD@data)[14]="SOC_BAU_20_max"
colnames(FORWARD@data)[15]="Med_Scen_min"
colnames(FORWARD@data)[16]="Med_Scen_max"
colnames(FORWARD@data)[17]="SOC_t0_min"
colnames(FORWARD@data)[18]="SOC_t0_max"
```

The 'euler' method can give some out of range results in some points, under specific combinations of climatic, soil and NPP variables. To avoid including those points in the maps we will remove any "out of range" value.

```
# Eliminate values out of range
FORWARD@data$SOC_BAU_20[FORWARD@data$SOC_BAU_20 <
    0] <- NA
FORWARD@data$Low_Scenario[FORWARD@data$Low_Scenario <
    0] <- NA
FORWARD@data$Med_Scenario[FORWARD@data$Med_Scenario <
    0] <- NA
FORWARD@data$High_Scenario[FORWARD@data$High_Scenario <
    0] <- NA
FORWARD@data$Med_Scen_min[FORWARD@data$Med_Scen_min <
    0] <- NA
FORWARD@data$Med_Scen_max[FORWARD@data$Med_Scen_max <
    0] <- NA

FORWARD@data$SOC_BAU_20[FORWARD@data$SOC_BAU_20 >
    300] <- NA
FORWARD@data$Low_Scenario[FORWARD@data$Low_Scenario >
    300] <- NA
FORWARD@data$Med_Scenario[FORWARD@data$Med_Scenario >
    300] <- NA
FORWARD@data$High_Scenario[FORWARD@data$High_Scenario >
```

```
    300] <- NA
FORWARD@data$Med_Scen_min[FORWARD@data$Med_Scen_min >
    300] <- NA
FORWARD@data$Med_Scen_max[FORWARD@data$Med_Scen_max >
    300] <- NA
```

Then, we will run the rest of the script, set the working directory, calculate the uncertainties and set the name of the output vector layer.

```
# Set the working directory

setwd(WD_OUT)

# UNCERTAINTIES

UNC_SOC<-((FORWARD@data$SOC_BAU_20_max-
             FORWARD@data$SOC_BAU_20_min)/
            (2*FORWARD@data$SOC_BAU_20))*100

UNC_t0<-((FORWARD@data$SOC_t0_max-FORWARD@data$SOC_t0_min)/
           (2*FORWARD@data$SOC_t0))*100

UNC_SSM<-((FORWARD@data$Med_Scen_max-FORWARD@data$Med_Scen_min)/
            (2*FORWARD@data$Med_Scenario))*100

FORWARD[[19]]<-UNC_SOC
FORWARD[[20]]<-UNC_t0
FORWARD[[21]]<-UNC_SSM

colnames(FORWARD@data)[19]="UNC_BAU"
colnames(FORWARD@data)[20]="UNC_t0"
colnames(FORWARD@data)[21]="UNC_SSM"

# SAVE the Points (shapefile)

writeOGR(FORWARD, ".", "FORWARD_County_AOI_parallel_test",
         driver="ESRI Shapefile",overwrite=TRUE)
```

Chapter 11

Stage 3: map generation

Once the model is run through the three proposed phases, we have all the information required for generating the maps. We need to transform the output vector to raster layers. We will obtain the SOC stocks after 20 years of SSM practices for the three scenarios (low, medium and high carbon inputs increments), and SOC stocks under the business as usual scenario (no carbon input increment). We will estimate four absolute carbon sequestration rates (considering the 2018 or 2020 SOC as a baseline), and three relative carbon sequestration rates (considering the SOC stocks under the business as usual as the baseline).

11.1 Script number 16: "Points_to_Raster.R"

We will use script number 16 to transform the output vector from script number 15 to raster layers. The inputs for this script are the output vector from script 15, the FAO SOC layer and the country boundary polygon. The outputs of the script number 16 are the SOC stocks for the future scenarios (20 years): BAU, low, medium and high carbon inputs, three relative sequestration rates (SOC stock SSM scenario - BAU scenario)/20 , and four absolute sequestration rates: (SOC stock SSM or BAU scenario - SOC stocks 2018/20)/20.

Target points to raster products. Inputs and outputs.

Input	Output
COUNTRY-POLYGON.shp (ROI) FORWARD_BAU_3E_20YEARS_ [code_country].shp (from script number 15) SOC_MAP_[country_code].tif (from script number 10)	3 raster scenarios (low, medium, and high) 1 raster BAU (business as usual) 4 absolute rates (scenario - 2018 SOC)/20 3 absolute differences (Scenario - 2018SOC) 3 relative rates (scenario - BAU) 3 relative differences (scenario -BAU) Uncertainty maps (BAU, SSM, SOC2018, UNC for absolute and relative rates)

We will open the script “Points_to_Raster.R” and load the required packages; then set the directories of the required files: Forward outputs, SOC, AOI, and the outputs maps folder.

```
rm(list=ls())
library(raster)
library(rgdal)
WD_F<-("C:/TRAINING_MATERIALS/OUTPUTS/3_FOWARD")
WD_SOC<-("C:/TRAINING_MATERIALS/INPUTS/SOC_MAP")
WD_AOI<-("C:/TRAINING_MATERIALS/INPUTS/AOI_POLYGON")
WD_MAPS<-("C:/TRAINING_MATERIALS/OUTPUTS/4_MAPS")
```

Now we will define the name of the output area of interest / country or region. In this example, “Pergamino”.

```
#Define the name of the Country ("ISO3CountryCode")
name<-"Pergamino"
```

Then open the layers:

```
#Open FORWARD vector
setwd(WD_F)
FORWARD<-readOGR("FOWARD_AOI_parallel.shp")
#Open SOC MAP (master layer)
setwd(WD_SOC)
SOC_MAP<-raster("SOC_MAP_AOI.tif")
```

```
#Creates emtpy raster
empty_raster<-SOC_MAP*0
# Open the country vector boundaries
setwd(WD_AOI)
Country<-readOGR("Departamento_Pergamino.shp")
# Cut the raster with the country vector
Country_raster<-crop(empty_raster,Country)
# Replace Na values for zero values
FORWARD@data[is.na(FORWARD@data)] <- 0
```

Next, we will transform the vector points from the FORWARD phase of the model to raster files using the “rasterize” function.

```
# Points to Raster BAU
setwd(WD_MAPS)
Country_BAU_2040_Map<-rasterize(FORWARD, Country_raster,
                  FORWARD$SOC_BAU_20, updateValue='all')
writeRaster(Country_BAU_2040_Map,
  filename=paste0(name,"_GSOCseq_finalSOC_BAU_Map030"),
  format="GTiff", overwrite = TRUE)
# Points to Raster Low Scenario
Country_Lwr_2040_Map<-rasterize(FORWARD, Country_raster,
                  FORWARD$Lw_Sc, updateValue='all')
writeRaster(Country_Lwr_2040_Map,
  filename=paste0(name,"_GSOCseq_finalSOC_SSM1_Map030"),
  format="GTiff", overwrite = TRUE)
# Points to Raster Med Scenario
Country_Med_2040_Map<-rasterize(FORWARD, Country_raster,
                  FORWARD$Md_Sc, updateValue='all')
writeRaster(Country_Med_2040_Map,
  filename=paste0(name,"_GSOCseq_finalSOC_SSM2_Map030"),
  format="GTiff", overwrite = TRUE)
# Points to Raster High Scenario
Country_Hgh_2040_Map<-rasterize(FORWARD, Country_raster,
                  FORWARD$Hgh_S, updateValue='all')
writeRaster(Country_Hgh_2040_Map,
  filename=paste0(name,"_GSOCseq_finalSOC_SSM3_Map030"),
  format="GTiff", overwrite = TRUE)
```

```
# Points to Raster initial SOC (t0) 2018/2020
Country_SOC_2018_Map<-rasterize(FORWARD, Country_raster,
                  FORWARD$SOC_t0, updateValue='all')
writeRaster(Country_SOC_2018_Map,
  filename=paste0(name,"_GSOCseq_T0_Map030"),
  format="GTiff", overwrite = TRUE)
```

Now, we will calculate the absolute differences and the absolute rates (SSM - SOC 2018).

```
# Difference BAU 2040 - SOC 2018
Diff_BAU_SOC_2018<-Country_BAU_2040_Map-Country_SOC_2018_Map
writeRaster(Diff_BAU_SOC_2018,filename=paste0(name,
"_GSOCseq_AbsDiff_BAU_Map030"),format="GTiff", overwrite = TRUE)
writeRaster(Diff_BAU_SOC_2018/20,filename=paste0(name,
"_GSOCseq_ASR_BAU_Map030"),format="GTiff", overwrite = TRUE)
# Difference Low Scenario - SOC 2018
Diff_Lw_SOC_2018<-Country_Lwr_2040_Map-Country_SOC_2018_Map
writeRaster(Diff_Lw_SOC_2018,filename=paste0(name,
"_GSOCseq_AbsDiff_SSM1_Map030"),format="GTiff", overwrite = TRUE)
writeRaster(Diff_Lw_SOC_2018/20,filename=paste0(name,
"_GSOCseq_ASR_SSM1_Map030"),format="GTiff", overwrite = TRUE)
# Difference Med Scenario - SOC 2018
Diff_Md_SOC_2018<-Country_Med_2040_Map-Country_SOC_2018_Map
writeRaster(Diff_Md_SOC_2018,filename=paste0(name,
"_GSOCseq_AbsDiff_SSM2_Map030"),format="GTiff", overwrite = TRUE)
writeRaster(Diff_Md_SOC_2018/20,filename=paste0(name,
"_GSOCseq_ASR_SSM2_Map030"),format="GTiff", overwrite = TRUE)
# Difference High Scenario - SOC 2018
Diff_Hg_SOC_2018<-Country_Hgh_2040_Map-Country_SOC_2018_Map
writeRaster(Diff_Hg_SOC_2018,filename=paste0(name,
"_GSOCseq_AbsDiff_SSM3_Map030"),format="GTiff", overwrite = TRUE)
writeRaster(Diff_Hg_SOC_2018/20,filename=paste0(name,
"_GSOCseq_ASR_SSM3_Map030"),format="GTiff", overwrite = TRUE)
```

Next, we will calculate the relative differences and rates (SSM - SOC BAU).

```
# Difference Low Scenario - BAU 2040
Diff_Lw_BAU_2040<-Country_Lwr_2040_Map-Country_BAU_2040_Map
writeRaster(Diff_Lw_BAU_2040,filename=paste0(name,
      "_GSOCseq_RelDiff_SSM1_Map030"),
      format="GTiff", overwrite = TRUE)
writeRaster(Diff_Lw_BAU_2040/20,filename=paste0(name,
      "_GSOCseq_RSR_SSM1_Map030"),
      format="GTiff", overwrite = TRUE)
# Difference Med Scenario - BAU 2040
Diff_Md_BAU_2040<-Country_Med_2040_Map-Country_BAU_2040_Map
writeRaster(Diff_Md_BAU_2040,filename=paste0(name,
      "_GSOCseq_RelDiff_SSM2_Map030"),
      format="GTiff", overwrite = TRUE)
writeRaster(Diff_Md_BAU_2040/20,filename=paste0(name,
      "_GSOCseq_RSR_SSM2_Map030"),
      format="GTiff", overwrite = TRUE)
# Difference High Scenario - BAU 2040
Diff_Hg_BAU_2040<-Country_Hgh_2040_Map-Country_BAU_2040_Map
writeRaster(Diff_Hg_BAU_2040,filename=paste0(name,
     "_GSOCseq_RelDiff_SSM3_Map030"),
     format="GTiff", overwrite = TRUE)
writeRaster(Diff_Hg_BAU_2040/20,filename=paste0(name,
      "_GSOCseq_RSR_SSM3_Map030"),
      format="GTiff", overwrite = TRUE)
```

Now, we will rasterize the values of the uncertainties of SOC BAU, SOC 2018 and one SSM (one for the three scenarios).

```
# Uncertainties SOC 2018
UNC_2018<-rasterize(FORWARD, Country_raster,
                    FORWARD$UNC_0, updateValue='all')
writeRaster(UNC_2018,filename=paste0(name,
      "_GSOCseq_T0_UncertaintyMap030"),
      format="GTiff", overwrite = TRUE)
# Uncertainties SOC BAU 2038
UNC_BAU<-rasterize(FORWARD, Country_raster,
                    FORWARD$UNC_B, updateValue='all')
writeRaster(UNC_BAU,filename=paste0(name,
```

```
      "_GSOCseq_BAU_UncertaintyMap030"),
      format="GTiff", overwrite = TRUE)
# Uncertainties SOC SSM
UNC_SSM<-rasterize(FORWARD, Country_raster,
                   FORWARD$UNC_S, updateValue='all')
writeRaster(UNC_SSM,filename=paste0(name,
      "_GSOCseq_SSM_UncertaintyMap030"),
      format="GTiff", overwrite = TRUE)
```

Now we will calculate the uncertainties for the absolute rates.

```
# Uncertainties for the Absolute difference SSM_ - SOC2018
UNC_abs_rate_BAU<-sqrt((FORWARD$UNC_B*FORWARD$SOC_BAU_20)^2+
  (FORWARD$UNC_0*FORWARD$SOC_t0)^2)/
  abs(FORWARD$SOC_t0+FORWARD$SOC_BAU_20)
UNC_abs_rate_BAU_Map<-rasterize(FORWARD,Country_raster,
                      UNC_abs_rate_BAU, updateValue='all')
writeRaster(UNC_abs_rate_BAU_Map,filename=paste0(name,
        "_GSOCseq_ASR_BAU_UncertaintyMap030"),
        format="GTiff", overwrite = TRUE)
UNC_abs_rate_Lw<-sqrt((FORWARD$UNC_S*FORWARD$Lw_Sc)^2 +
      (FORWARD$UNC_0*FORWARD$SOC_t0)^2)/
        abs(FORWARD$SOC_t0+FORWARD$Lw_Sc)
UNC_abs_rate_Lw_Map<-rasterize(FORWARD,Country_raster,
                           UNC_abs_rate_Lw, updateValue='all')
writeRaster(UNC_abs_rate_Lw_Map,filename=paste0(name,
          "_GSOCseq_ASR_SSM1_UncertaintyMap030"),
          format="GTiff", overwrite = TRUE)

UNC_abs_rate_Md<-sqrt((FORWARD$UNC_S*FORWARD$Md_Sc)^2+
    (FORWARD$UNC_0*FORWARD$SOC_t0)^2)/
      abs(FORWARD$SOC_t0+FORWARD$Md_Sc)
UNC_abs_rate_Md_Map<-rasterize(FORWARD,Country_raster,
               UNC_abs_rate_Md, updateValue='all')
writeRaster(UNC_abs_rate_Md_Map,filename=paste0(name,
          "_GSOCseq_ASR_SSM2_UncertaintyMap030"),
          format="GTiff", overwrite = TRUE)
UNC_abs_rate_Hg<-sqrt((FORWARD$UNC_S*FORWARD$Hgh_S)^2 +
```

```
      (FORWARD$UNC_0*FORWARD$SOC_t0)^2)/
        abs(FORWARD$SOC_t0+FORWARD$Hgh_S)
UNC_abs_rate_Hg_Map<-rasterize(FORWARD,Country_raster,
                            UNC_abs_rate_Hg, updateValue='all')
writeRaster(UNC_abs_rate_Hg_Map,filename=paste0(name,
          "_GSOCseq_ASR_SSM3_UncertaintyMap030"),
          format="GTiff", overwrite = TRUE)
```

Now we will calculate the uncertainties for the relative rates.

```
# Uncertainties for the Relative difference  SSM_ - SOCBAU
UNC_Rel_rate_Lw<-sqrt((FORWARD$UNC_S*FORWARD$Lw_Sc)^2+
  (FORWARD$UNC_B*FORWARD$SOC_BAU_20)^2)/
  abs(FORWARD$SOC_BAU_20+FORWARD$Lw_Sc)
UNC_Rel_rate_Lw_Map<-rasterize(FORWARD,Country_raster,
                       UNC_Rel_rate_Lw, updateValue='all')
writeRaster(UNC_Rel_rate_Lw_Map,filename=paste0(name,
       "_GSOCseq_RSR_SSM1_UncertaintyMap030"),
       format="GTiff", overwrite = TRUE)
UNC_Rel_rate_Md<-sqrt((FORWARD$UNC_S*FORWARD$Md_Sc)^2+
  (FORWARD$UNC_B*FORWARD$SOC_BAU_20)^2)/
  abs(FORWARD$SOC_BAU_20+FORWARD$Md_Sc)
UNC_Rel_rate_Md_Map<-rasterize(FORWARD,Country_raster,
                       UNC_Rel_rate_Md, updateValue='all')
writeRaster(UNC_Rel_rate_Md_Map,filename=paste0(name,
      "_GSOCseq_RSR_SSM2_UncertaintyMap030"),
      format="GTiff", overwrite = TRUE)

UNC_Rel_rate_Hg<-sqrt((FORWARD$UNC_S*FORWARD$Hgh_S)^2+
  (FORWARD$UNC_B*FORWARD$SOC_BAU_20)^2)/
  abs(FORWARD$SOC_BAU_20+FORWARD$Hgh_S)
UNC_Rel_rate_Hg_Map<-rasterize(FORWARD,Country_raster,
                       UNC_Rel_rate_Hg, updateValue='all')
writeRaster(UNC_Rel_rate_Hg_Map,filename=paste0(name,
      "_GSOCseq_RSR_SSM3_UncertaintyMap030"),
      format="GTiff", overwrite = TRUE)
```

Chapter 12

Uncertainty and validation

Ideally, model prediction uncertainty provided in the GSOCseq map should include all sources of uncertainty that affect predictions, including model structural uncertainty, model parameters' and input data uncertainties. As a minimum, uncertainty should include input data uncertainties (e.g. Morais *et al.*, 2019). There are different methods to estimate uncertainties in the results. Monte Carlo methods, that draw random values from the probability distribution functions for inputs and parameters, are an efficient way to estimate the whole uncertainty of the modeled estimation (Ogle *et al.*, 2010; FAO, 2019b; Morais *et al.*, 2019). In Monte Carlo simulation methods, parameter values of the model and input data (e.g. mean temperature, clay content, carbon inputs) shall be randomly chosen from hypothetical normal distributions with mean equal to the parameter value and the measured standard error around that mean. Once all the different parameter values for the model are generated from the hypothetical distributions, a model run shall be made. This process is to be repeated 100 or more times to produce a mean model prediction with a 95 percent confidence interval. The Monte Carlo simulation would generate an expected value of SOC stocks for the different scenarios and a 95 percent confidence interval. Uncertainty (U) shall be expressed as a percentage: half of the 95 percent confidence interval divided by the mean (Ogle *et al.*, 2010). Thus, uncertainty can be estimated for each simulated scenario as:

$$U\% = \frac{100 \times (ULCI - LLC)}{2 \times SOC_{av}} \tag{12.1}$$

where UL corresponds to the upper limit of the 95 percent confidence interval of the estimated SOC at the end of the simulation (in tC/ha), LL corresponds to the lower limit of the 95 percent confidence interval of the estimated SOC at the end of the simulation (in tC/ha); and SOC_{av} the average of the estimated SOC at the end of the simulation (tC/ha), after 20 years of the forward modelling, for each scenario. To estimate uncertainties of the sequestration rates (uncertain quantities are combined by subtraction, e.g. Δ SOC = Stocks SSM – SOC stocks BAU), the uncertainty can be expressed in percentage terms was estimated by the following equation (IPCC, 2019):

$$Ut = \sqrt{((U1X1)^2 + ... + (UnXn)^2)}/|X1 + ... + Xn| \quad (12.2)$$

where Ut is the percentage uncertainty in the subtraction of the quantities (half the 95 percent confidence interval divided by the total, i.e. mean, and expressed as a percentage), $x1$ – n represent the quantities to be combined (e.g. Stocks SSM and SOC stocks BAU at the end of the forward simulation), and $U1$ to n is the percentage uncertainties associated with each of the quantities (as estimated from equation 12.1). However, Monte Carlo and related simulations (e.g. Markov Chain–Monte Carlo method, as in Hararuk *et al.*, 2014; GLUE method, as in Salazar *et al.*, 2011) usually require considerable computational capacity and may be time demanding, especially for long spin–up runs (>500 years), as multiple (>100) runs should be made for each modeling unit. An alternative is to calculate uncertainties of the input data considering minimum and maximum values (corresponding to the limits of a 95 percent confidence interval) of a set of predefined input parameters, considered to have the greatest influence in RothC modeling results (initial SOC, Carbon inputs, and soil and climatic variables). Thus, uncertainties can be estimated for each modeling unit and for each scenario by estimating first the minimum and maximum SOC simulated values (similarly to VCS, 2012) using a predefined arrangement of inputs:

$$SOCmax = Model(SOC_{FAO\ max}, Ci_{max}, Temp_{min}, Pp_{max}, Clay_{max}) \quad (12.3)$$

$$SOCmin = Model(SOC_{FAO\ min}, Ci_{min}, Temp_{max}, Pp_{min}, Clay_{min}) \quad (12.4)$$

where $SOC_{FAO\ min}$ and $SOC_{FAO\ max}$ are respectively the minimum and maximum value for the simulated SOC stocks; $SOC_{FAO\ min}$ and $SOC_{FAO\ max}$ are

the minimum and maximum value for the initial SOC GSCOCmap stocks estimated at the 95 percent confidence interval are the Ci_{min} and Ci_{max} are respectively the minimum and the maximum value for the annual carbon inputs estimated at the 95 percent confidence interval; $Temp_{min}$ and $Temp_{max}$ are respectively the minimum and maximum value for the average monthly air temperature estimated at the 95 percent confidence interval; Pp_{min} and Pp_{max} are respectively the minimum and maximum value for the average monthly precipitation estimated at the 95 percent confidence interval; and Clay min and Clay max are respectively the minimum and maximum value for the soil clay content (0–30 cm) estimated at the 95 percent confidence interval. The arrangement of variables to generate minimum and maximum SOC stocks are to be generated considering the effects of each variable on NPP, decomposition rates, and overall carbon dynamics (Chapters 4 and 5). If information is available, the minimum and maximum value of each parameter (C input, Temp, Pp and Clay) that define de 95 percent confidence interval can be estimated from its variation and mean value, assuming that values of the parameter are normally distributed about the mean:

$$P_{min} = Xp - 1.96 \times SEp \tag{12.5}$$

$$P_{max} = Xp - 1.96 \times SEp \tag{12.6}$$

where P_{min} and P_{max} are respectively the minimum and maximum value for parameter P (C input, Temp, Pp or Clay) estimated at the 95 percent confidence interval; Xp is the average value of that parameter; and SEp is the standard error of the mean of that parameter. Uncertainties already generated in the latest GSOCmap can be used to obtain the min and max SOC FAO values. Uncertainties in C inputs and thus Ci max and min can be estimated from available data (e.g. meta–analysis). Temp max and Temp and PP max and PPmin can be estimated from the average monthly values and confidence intervals of the climatic series to be modeled. Uncertainties in clay contents can be directly obtained from SoilGrids (https://soilgrids.org/) if the ISRIC database is to be used for the clay content layers. If no estimate of clay variation is available for the used database, Clay max and clay min can be determined from clay content variation within the 1km x 1km grid cells (i.e. considering the values from 250 m x 250 m resolution grids). If no estimate of the SE or CI is available for these parameters, a maximum and minimum value can be estimated for these

parameters, using general uncertainty coefficients, as those reported from global modelling exercises by Gottschalk *et al.* (2007) and Hastings *et al.* (2010). Average uncertainties for these parameters are summarized in Table 12.1. Input data layers (SOC FAO, temperature, precipitation, clay content, and C inputs) need to be re–prepared for the different modelling phases considering the maximum and minimum values for each data input outlined in equations 12.3 and 12.4. General uncertainties in SOC sequestration can be then estimated for all scenarios. The model is to be run 2 two more times for each modelling unit and scenario in the different modelling phases: once using the selection of values to obtain a maximum expected SOC (eq 12.2), and once using the selection of input values to obtain a minimum expected SOC change (eq 12.3). SOC stocks are then modelled for each modelling unit. Uncertainties can be then expressed in percent as in equation 12.1 for each scenario (considering the average SOC values as the ones obtained in the first modelling run for each scenario):

$$U\% = \frac{100 \times (SOC_{max} - SOC_{min})}{2 \times SOC} \tag{12.7}$$

where SOC_{max} corresponds to the upper limit of simulated the SOC stocks (in tC/ha) at the end of the forward modelling phase using the combination of inputs in equation 12.3, SOC_{min} corresponds to the lower limit of the simulated SOC stocks (in tC/ha) at the end of the forward modelling phase using the combination of inputs in equation 12.4; and SOC the average simulated SOC stocks (tC/ha), for each scenario, after 20 years of the forward modeling phase. Uncertainties for the absolute and relative sequestration rates can be then estimated using equation 12.2, for each scenario.

Uncertainties of main parameters affecting SOC dynamics following Gottschalk et al.(2007) and Hastings et al. (2010).

Parameter	Uncertainty in the input	Minimum value	Maximum value
Temperature	± 2 %	Monthly Temp * 0.98	Monthly Temp * 1.02
Precipitation	± 5 %	Monthly PP * 0.95	Monthly PP * 1.05
Clay content	± 10 %	Clay * 0.90	Clay * 1.10
FAO SOC	± 20 %	SOC FAO *0.8	SOC FAO * 1.2
C input increase in SSM scenario	± 15 %	C eq * (SSM1 % increase - 15%)	C eq * (SSM % increase + 15%)

The model should be validated for the conditions in which it will be applied when possible. The use of models for prediction involves a series of problems for validation, as data required to quantify the accuracy of the estimates do not

yet exist. Nonetheless, predictive models can be validated if they explain past events (ex–post validation). If local results from different SSM practices on SOC stocks are available (a meta–analysis of local SSM practices can be conducted), and the collected activity data allow to perform simulations with these records, model–produced estimates shall be compared with the observed results. The RMSE shall be used to compare the divergence between model estimates and field observations. The RMSE can be expressed as:

$$RMSE = \frac{\sqrt{(\sum_{i=1}^{n}(0_i - P_i)^2)}}{n} \tag{12.8}$$

where Pi is the predicted (modelled) value, Oi is the observed value, n is the number of measured Relative $RMSE$ can be expressed as a percent of the observed mean. These results should be specified in the report accompanying the mapping product.

Chapter 13

Data sharing

13.1 General principles

Sharing of the national GSOCseq products will follow the same approach as other GSP global mapping products such as the Global Soil Organic Carbon Map (GSOCmap – 2017) and Global Map of Salt–Affected Soils (GSASmap – 2020). The GSP Secretariat will provide an online data submission facility. Data sharing is governed by the GSP Data Policy (http://www.fao.org/3/a-bs975e.pdf). When delivering national data to the GSP, countries need to ensure that:

1. The data shared contain the relevant soil information representative for the country.
2. The shared datasets contain the best available information for a given area and topic, however, subject to potential restrictions based on the institutions' or countries' data policy.
3. The data shared are reliable and quality controlled which means that the data have passed a technical routine to ensure data integrity, correctness, and completeness; errors and omissions are identified and, if possible, addressed.

Data providers will retain the ownership of national datasets. The final global dataset will be distributed under Creative Commons license.

13.2 Data formats and integrity

All GIS files should be delivered in correct format according to the Technical Specifications and Country Guidelines for the GSOCseq map. GIS files shall be delivered in GeoTIFF format with embedded additional georeferencing information, including spatial extent, coordinate reference system, resolution, no data values. Pixel values should contain numeric values of the target parameter, estimated for the topsoil layer (0–30 cm depth). World Geodetic System 1984 (WGS84) geographic (latitude/longitude) projection should be preferred for all submitted maps. The map shall be produced at regular fixed horizontal dimensions of 30 by 30 arc–seconds grid (approximately only 1 x 1 km) at the equator. Countries will be expected to deliver their datasets using the standard 30 arc–second grids prepared and shared by the GSP. Each data layer should be provided as a single continuous surface. Data providers should not attempt to mask out the excluded areas from the grid (e.g. saline soils, organic soils, wetlands). The GSP Secretariat will mask excluded areas using standard spatialized layers. Values in the excluded grid cells will be identified as no data (NA) in the final global product.

13.3 Data correctness

Each country will be responsible for carrying out basic Quality Assurance/Quality Control (QA/QC) of all data before providing it to the GSP Secretariat. Quality Assurance can be described as the process of preventing errors from entering the datasets, while Quality Control can be described as the process of identifying and correcting existing errors in the datasets. All datasets should be checked for:

- Erroneous or unrealistic predictions
- Spatial errors (extent, projection)
- Units (annual tC/ha)
- Completeness and integrity of data
- Consistency with data shown in any accompanying documents (such as reports or drawings)
- Validation (if possible) and consistency of reported validation results with the provided data.

Final QA/QC for the global datasets will be facilitated by the GSP Secretariat through its technical networks (INSII, P4WG, and Intergovernmental Technical Panel on Soils (ITPS) will give final clearance to the global dataset prior to public release).

13.4 Data completeness

A complete set of mandatory data layers should be provided. This complete set of SOC sequestration maps includes country–level predicted topsoil (0–30 cm) SOC stocks and mean annual sequestration rates after the implementation of SSM practices, for a 20–year period, estimated with the spatialized version of the RothC carbon model. Each pixel shall contain:

1. Annual absolute average (vs. initial SOC stock) sequestration rates for 20 years (tC/ha), for BAU, SSM1, SSM2 and SSM3 scenarios
2. Annual mean relative (vs. business as usual) sequestration rates for 20 years (tC/ha), for SSM1, SSM2 and SSM3 scenarios
3. Initial SOC stocks (tC/ha)
4. SOC sequestration uncertainty maps (absolute and relative sequestration rates) for 3 SSM Scenarios (in percent)

The set of data layers should be supplemented by a country report (electronic document) according to the submission form provided by GSP–FAO.

Country members are also encouraged to deliver the following products and supplementary data (optional):

- final SOC stocks at 2040 (tC/ha), for BAU, SSM1, SSM2 and SSM3 scenarios;
- SOC sequestration maps using alternative modeling procedures: Includes country–level topsoil (0–30 cm) predicted SOC stocks and mean annual sequestration rates after implementation of SSM practices, for a 20–year period (2020–2040), estimated using;
- alternative SSM scenarios (e.g. alternative C percent increase based on local data analysis);
- alternative local preferred, process–oriented and peer–reviewed models (e.g. CENTURY/DAYCENT, DNDC, YASSO, ICBM, or their derived models);

- meta–analysis on the local impact of SSM management practices on SOC sequestration;
- validation dataset (results from predicted vs. observed SOC stocks/SOC sequestration rates from meta–analysis and RMSE; validation dataset in table format; shapefile/points if georeferenced data is available).

13.5 Other modes of data sharing

Apart from sharing national data with the GSP for the development of the Global Soil Organic Carbon Sequestration Potential Map, countries are encouraged to share and use their maps in a number of ways, including:

- scientific publications;
- inclusion in national soil information systems;
- using the maps for the development of national policy regarding; sustainable and country–smart agriculture;
- using the maps for identifying key areas for governmental, non–governmental, commercial, national or international projects, related to promoting sustainable soil management and carbon sequestration;
- using the maps for awareness raising on the topics of climate change and soil health;
- using the maps for educational purposes at schools and universities for subjects related to geography, climate, soils, agriculture, etc.

GSP encourages data providers to apply open licensing and free data sharing of their products to ensure broad distribution of knowledge and high impact towards sustainable management of our soils.

Chapter 14

Limitations, knowledge gaps and way forward

It is of most importance to identify which regions, environments and production systems present greater potential to increase SOC stocks and mitigate GHG emissions and establish priorities for research and implementation of public policies. In this document, we provided an approach and the procedures to produce digital SOC sequestration maps using soil legacy data, process oriented SOC models and modern techniques of digital soil mapping, that would allow covering as many conditions and productive systems worldwide as possible, in a relatively simple, transparent and standardized way, without complex configuration and computational capacities. Nonetheless, the estimation of SOC sequestration potential in a harmonized way among countries, regions, and productive systems is not an easy task and there are different contentious issues and limitations that must be outlined.

Firstly, agricultural lands (croplands and grazing lands) are selected as target areas to estimate SOC sequestration potential in this first instance, since they are managed at least on a yearly basis, and management practices could be used to increase soil organic carbon content. These lands have been identified as the options with greater potential to accumulate SOC and mitigate GHG emissions through improved management practices (Smith *et al.*, 2008; Lal *et al.*, 2018). Furthermore, most of the information regarding the SOC dynamics has been developed in these productive systems, and most SOC carbon models

have been successfully tested under these conditions. Countries can nevertheless assess SOC sequestration potential of different land uses and deliver additional maps including other land use (other than agricultural lands). Future versions of the GSOCseq map may include other land uses, depending on national demands.

Secondly, most SOC models are parameterized under land use, land management, soil or climatic regions. Ideally, SOC models should account for all major SOC–controlling factors, such as soil mineralogy, climate conditions, litter quality, biota activity, land use and management. These factors have extremely complex interactions, and separate analysis of controls could limit predictions of their effects on SOC (Falloon and Smith, 2009). Even the full multidimensional development of a single element of a model can rarely, if ever, be predicted precisely, and the actual consequence is that it is impossible to create "universal" models (Sinclair and Seligman, 1996). The review by Campbell and Paustian (2015) emphasizes the fact that among the different known process–oriented models used to estimate SOC changes, no one clearly outperforms the others. However, in order to obtain consistent and harmonized results, and allow comparisons between countries and regions, due to potential differences in computational, technical capacities and data availability, the use of RothC as a standard 'process–oriented' SOC model, following the proposed methodology, is requested as a first step. Nevertheless, users are encouraged to provide supplementary alternative maps developed using alternative preferred SOC models and/or methods or approaches to estimate C inputs and compare results with the proposed methodology. The use of a multi–model ensemble approach (e.g. Riggers et al, 2019; Lehtonen *et al.*, 2020) with selected models is intended for future versions of the GSOCseq map. Moreover, the SoilR package (Sierra *et al.*, 2012) used in the current approach already includes other SOC models like CENTURY and ICBM, that can be used to estimate results using a multi–model ensemble approach.

It must be also outlined that at some level of analysis all known process–oriented SOC models including RothC (see Chapter 2; see FAO, 2019), include empirical functions, so they are expected to perform best when operating in situations similar to those for which they were originally parameterized, which tend to be croplands and grasslands from the temperate zone (Jenkinson *et al.*, 1990; Petri *et al.*, 2010). There is relatively less available data of the performance of SOC models under tropical and arid conditions. Current SOC models, including RothC, may be limited in their applicability to these systems, due to differences in soil fauna and their effects on SOC dynamics, the much faster turnover of slow and passive SOM, different temperature and moisture relationships with micro-

bial activity, and differences in mineralogy (Shang and Tiessen, 1998; Tiessen *et al.*, 1998) and solution chemistry (Parton *et al.*, 1989) in tropical soils, or water dynamics under arid environments (Farina *et al.*, 2013). The inability to account for cation availability or aluminium (Al) toxicity may also limit SOC model predictions (Parton *et al.*, 1989; Shang and Tiessen, 1998). Most well–known models may be limited by failing to account for pH effects on soil carbon turnover (Jenkinson, 1988; 1996; Falloon and Smith, 2009). Soil organic carbon models generally predict faster carbon turnover than the ones observed in very acid soils (Motavalli *et al.*, 1995), and few models can predict SOC changes in allophanic soils or soils developed on recent volcanic ash (Jenkinson *et al.*, 1991; Motavalli *et al.*, 1995; Falloon *et al.*, 1998; Falloon and Smith, 2000; Falloon and Smith., 2009).

Additionally, RothC does not accurately simulate SOC dynamics in waterlogged soils such as paddy rice. Countries are encouraged to use local adaptations or modified versions of the RothC model which have shown to improve estimations under the above mentioned conditions (e.g. Parshotam and Hewitt, 1995; Saggar *et al.*, 1996 for volcanic soils; Shirato *et al.*, 2004; Shirato and Yokozawa, 2005 for paddy rice; and ROTHC10 developed by Farina *et al.*, 2013 in arid conditions). Local adaptations should be implemented following the general procedures and input data described in Chapter 5 and 6 to obtain consistent results, and/or to use their preferred model (e.g. Gilhespy *et al.*, 2014) under these conditions, and deliver additional maps to contrast results. Further developments of the GSOCseq will include specific and standardized methods for SOC estimations in paddy rice and other specific conditions.

The proposed approach estimates SOC changes in the first 0–30 cm. Although SOC at deeper soil layers is responsive to land management changes (e.g. Follett *et al.*, 2013; Poeplau and Don, 2013; Schmer *et al.*, 2014), the 0–30 cm is selected because: it is most responsive to land management changes; allows the use of GSOCmap as a baseline for SOC stocks; allows for better harmonization with national greenhouse gas inventories, and allows validation of selected models with available ground data (mostly generated at 0–30cm depth). New models and adaptations of known models have been developed to account for SOC dynamics in deep layers with different approaches (see Campbell and Paustian, 2015). For example, the DAYCENT model was modified to simulate deeper soil C dynamics by slowing SOC pool turnover and increasing allocation to passive soil C, without separating soil layers (Wieder *et al.*, 2014). Jenkinson and Coleman (2008) modified RothC to RothPC–1 to predict the turnover of organic C in subsoils up to 1 m of depth using multiple layers and introduced two ad-

ditional parameters, one that transports organic C down the soil profile by an advective process, and one that reduces decomposition processes of SOC with depth. However, there is still a strong necessity for additional data to confirm or refute hypotheses suggested by the different modelling approaches of SOC in deep layers (Campbell and Paustian, 2015). As new information is generated, future versions of the GSOC and GSOCseq maps will be able to incorporate SOC stocks and SOC changes at deeper layers. There is also still need for a better understanding of spatial heterogeneity in SOC in the landscape and for a better prediction of potential changes in SOC dynamics on the landscape scale (Stockmann *et al.*, 2013). Differences in drainage that may be linked to landscape position are often not accounted for in SOC models (Falloon and Smith, 2009). In this sense, three gaps in knowledge have been identified (Stockmann *et al.*, 2013): (1) the development of optimal, but still simple, 3–dimensional representations of landscapes (vertically and horizontally), (2) the implementation of functional interactions and SOC transfers (i.e. the redistribution of SOC to different parts in the landscape due to erosion, transport and deposition) and (3) the availability of adequate datasets for model validations (especially the representation of fluxes between different landscape elements).

It should be also outlined that the temperature is expected to increase in the next 20–50 years, especially after 2050 (IPCC, 2018) and this may impact SOC dynamics. The proposed approach considers a 2000–2020 climate average for SOC projections after 2020. Using a 20–year average removes the year–to–year variation. However, there is no consensus over which climate projections are the most appropriate for 2020–2040, and prior agreement between countries is required. The proposed methodology allows climate change scenarios for longer–term projections to be incorporated in future versions.

It should be also noted that a very wide range of management practices are currently being implemented and can potentially be introduced into the world's agricultural systems, depending on climatic, soil, socio–cultural and economic conditions. In turn, different SSM C–oriented practices are often combined, making it difficult to dissociate their effects on SOC dynamics. Thus, as a first step, and to harmonize the results on a global map, and because soil carbon turnover models are the most sensitive to carbon inputs (FAO, 2019), this manual proposed to group SSM practices into three scenarios as a standard method, based on their expected relative effects on C inputs compared to business as usual management: Low, Medium and High increase in C inputs (referred as SSM1, SSM2, and SSM3 scenarios). A 5 percent–10 percent–20 percent increase in C input is suggested as default values to test potential. This increase in C–

inputs will not always be possible where C–inputs are already high. On the other hand, this increase in C inputs can be low in regions or productive systems with current low C inputs. However, to obtain consistent and harmonized results, and allow comparisons between countries and regions the use of standard SSM scenarios and 5, 10, 20 percent is kindly requested in this first stage. National experts' opinion and local data are essential in order to accurately estimate or validate the target areas and carbon input levels for the different SSM scenarios in forthcoming versions. Countries are encouraged to provide supplementary alternative maps developed using alternative percent increases in C inputs or specific absolute increases in C inputs of specific SSM practices in the different agro–ecological regions and productive systems of the country, based on local knowledge or obtained from a literature search of local studies. We believe the comparison of results will greatly enrich the final product. The information generated by the different countries will allow us to select and model specific practices in forthcoming versions.

Finally, the precision of models relies heavily on the quality and quantity of data used in executing and validating them (FAO, 2019). Often, the datasets for running models are not collected for that specific purpose but are taken from previous or ongoing studies. In many cases the format and amount of data may be inappropriate for the models. There may be several potential pitfalls for the integration of data to calibrate, drive and evaluate a SOC model. Careful harmonization of datasets and input estimation methodologies is essential to obtain consistent results across regions and countries. Ideally, calibration and driving data should match the scale of the model simulation. However, data limitations will prompt the use of data of coarser resolution and/or mixing data of varying quality from different sources. The 1 km x 1 km scale for the GSOCseq is required in the final product to allow comparisons among countries. However, input datasets from different resolutions will be probably used, and this may introduce uncertainties (e.g. climate data that usually occurs at coarser resolutions). In the proposed approach, global sources are proposed (same resolution and quality) but countries are encouraged to develop and deliver SOCseq maps using the best available national climatic, soil, and land use data.

Data availability for model evaluation will also affect the assessment of model accuracy, as well as its ability to support hypothesis testing. Although there is a wealth of measured data from carefully monitored long–term agronomic experiments to evaluate SOC models, especially in the northern hemisphere and temperate climate conditions, there are comparatively few similar datasets from natural ecosystems (Falloon and Smith, 2009). The suite of datasets may then

become sources of uncertainty in SOC model predictions (Keenan *et al.*, 2011; Palosuo *et al.*, 2012). Datasets are also often difficult to identify or compare between SOC models, particularly in large–scale ecosystem or global analyses. Furthermore, soil carbon measurements from available experiments are rarely available in replicate and hence attributing uncertainty to these measurements, and ultimately confidence in SOC model predictions is limited (Falloon and Smith, 2003). Data availability to validate model performance will be a limitation for many countries.

A meta–analysis should be conducted on the basis of the latest available local and regional studies to estimate how agricultural practices affect average annual C inputs, SOC sequestration rates and SOC stocks. However, meta–analyses and comparisons have often suffered from datasets based on diverging definitions (e.g. concerning definitions of sample depth, baselines for comparisons, or the components of soil respiration), (Bahn *et al.*, 2012). We hope that this exercise, together with other GSP activities, will also be an opportunity for the different countries to establish long–term observatories that will allow us to monitor the effect of different management practices on SOC stocks under different environments, and this will in turn allow us to improve model estimations. We acknowledge that consistency among input datasets and results would be improved if there was only one actor involved in the entire process. However, it is of most importance that information is locally generated, involving local experts and institutions, building technical capacities in the process, as this will encourage countries to implement national and subnational policies, and to get involved in regional and global policies. Moreover, we believe that a 'bottom–up', country–driven approach, using country specific data and expert knowledge, is a fundamental step for iterative improvements. As it is the case of other GSP documents, this manual constitutes a living document, which will be continuously improved and refined after its use and implementation.

References

Abberton, M., Conant, R. & Batello, C. 2010. *Grassland carbon sequestration: Management, policy and economics.* Rome, FAO. 342pp. (also available at: https://www.fao.org/3/i1880e/i1880e.pdf)

Al-Adamat, R., Rawajfih, Z., Easter, M., Paustian, K., Coleman, K., Milne, E., Falloon, P., Powlson, D. & Batjes, N. 2007. Predicted soil organic carbon stocks and changes in Jordan between 2000 and 2030 made using the GEFSOC Modelling System. *Agriculture, Ecosystems & Environment*, 122(1): 35–45.

Allen, M. & Stocker, T. 2014. Impact of delay in reducing carbon dioxide emissions. *Nature Climate Change*, 4(1): 23–26.

Bahn, M., Kutsch, W. & Heinemeyer, A. 2012. Synthesis: emerging issues and challenges for an integrated understanding of soil carbon fluxes. In W. Kutsch, M. Bahn, & A. Heinemeyer, eds. *Soil Carbon Dynamics - an Integrated Methodology,* pp. 257-271. Cambridge, UK, Cambridge University Press.

Batjes, N. 1996. Total carbon and nitrogen in the soils of the world. *European Journal of Soil Science*, 47(2): 151–163.

Beek, J. & Frissel, M. 1973. *Simulation of nitrogen behaviour in soils,* 67pp. Wageningen, NL, Centre for Agricultural Publishing and Documentation.

Bolinder, M., Janzen, H., Gregorich, E., Angers, D. & VandenBygaart, A. 2007. An approach for estimating net primary productivity and annual carbon inputs to soil for common agricultural crops in Canada. *Agriculture, Ecosystems & Environment*, 118: 29-42.

Campbell, E. & Paustian, K. 2015. Current developments in soil organic

matter modeling and the expansion of model applications: a review. *Environmental Research Letters*, 10(12): 123004.

Clarholm, M & Bergström, L. 1989. *Ecology of Arable Land – Perspectives and Challenges: Proceeding of an International Symposium, 9-12 June 1987 Swedish University of Agricultural Sciences, Uppsala, Sweden,* 259pp. Dordrecht, NL, Kluwer Academic Publishers.

Coleman, K. & Jenkinson, D. 1996. RothC-26.3 - A Model for the turnover of carbon in soil. In D.S. Powlson, P. SMith, J. U. Smith, eds. *Evaluation of soil organic matter models,* pp. 237-246. Berlin, Heidelberg, Springer.

Easter, M., Paustian, K., Killian, K., Williams, S., Feng, T., Al-Adamat, R., Batjes, N. et al. 2007. The GEFSOC soil carbon modelling system: A tool for conducting regional-scale soil carbon inventories and assessing the impacts of land use change on soil carbon. *Agriculture, Ecosystems & Environment*, 122(1): 13–25.

Eggleston, H. S., Buendia, L., Miwa, K., Ngara, T. & Tanabe, K. 2006. *IPCC guidelines for national greenhouse gas inventories.* Kamiyamaguchi, Japan, Institute for Global Environmental Strategies.

Falloon, P., Smith, P., Smith, J., Szabo, J., Coleman, K. & Marshall, S. 1998. Regional estimates of carbon sequestration potential: linking the Rothamsted Carbon Model to GIS databases. *Biology and Fertility of Soils*, 27(3): 236-241.

Falloon, P. & Smith, P. 2003. Accounting for changes in soil carbon under the Kyoto Protocol: need for improved long-term data sets to reduce uncertainty in model projections. *Soil Use and Management*, 19(3): 265–269.

Falloon, P., Jones, C., Cerri, C., Al-Adamat, R., Kamoni, P., Bhattacharyya, T., Easter, M., Paustian, K., Killian, K., Coleman, K. & Milne, E. 2007. Climate change and its impact on soil and vegetation carbon storage in Kenya, Jordan, India and Brazil. *Agriculture, Ecosystems & Environment*, 122: 114-124.

FAO. 2017. *Voluntary Guidelines for Sustainable Soil Management,* 27pp. Rome, FAO. (also available at: https://www.fao.org/3/bl813e/bl813e.pdf)

FAO. 2021. *Global Soil Organic Carbon (GSOC) Map.* Rome, Food and Agriculture Organization of the United Nations. (available at: http://54.229.242.119/GSOCmap/)

Farina, R., Coleman, K. & Whitmore, A. 2013. Modification of the RothC model for simulations of soil organic C dynamics in dryland regions. *Geoderma*, 200-201: 18-30.

Farina, R., Marchetti, A., Francaviglia, R., Napoli, R. & Di Bene, C. 2017. Modeling regional soil C stocks and CO2 emissions under Mediterranean cropping systems and soil types. *Agriculture, Ecosystems & Environment*, 238: 128–141.

Follett, R., Kimble, J., Pruessner, E., Samson-Liebig, S. & Waltman, S. 2009. Soil Organic Carbon Stocks with Depth and Land Use at Various U.S. Sites. In R. Lala & R. F. Follett, eds. *Soil carbon sequestration and the greenhouse effect,* pp. 29–46. Madison, USA, Soil science society of America.

Franko, U. 1996. Modelling approaches of soil organic matter turnover within the CANDY system. In D.S. Powlson, P. SMith, J. U. Smith, eds. *Evaluation of soil organic matter models,* pp. 247–254. Berlin, Heidelberg, Springer.

Gilhespy, S., Anthony, S., Cardenas, L., Chadwick, D., Prado, A., Li, C., Misselbrook, T. et al. 2014. First 20 years of DNDC (DeNitrification DeComposition): Model evolution. *Ecological Modelling*, 292: 51–62.

Gottschalk, P., Smith, J., Wattenbach, M., Bellarby, J., Stehfest, E., Arnell, N., Osborn, T., Jones, C. & Smith, P. 2012. How will organic carbon stocks in mineral soils evolve under future climate? Global projections using RothC for a range of climate change scenarios. *Biogeosciences*, 9(8): 3151–3171.

Grace, P., Ladd, J., Robertson, G. & Gage, S. 2006. SOCRATES - A simple model for predicting long-term changes in soil organic carbon in terrestrial ecosystems. *Soil Biology and Biochemistry*, 38(5): 1172–1176.

Hadas, A., Parkin, T. & Stahl, P. 1998. Reduced CO 2 release from decomposing wheat straw under N-limiting conditions: simulation of carbon turnover. *European Journal of Soil Science,* 49: 487-494.

Hansen, S., Jensen, H., Nielsen, N. & Svendsen, H. 1991. Simulation of nitrogen dynamics and biomass production in winter wheat using the Danish simulation model DAISY. *Fertilizer research,* 27(2-3): 245–259.

Hengl, T., Jesus, J., Heuvelink, G., Ruiperez Gonzalez, M., Kilibarda, M., Blagotic, A., Shangguan, W. et al. 2017. SoilGrids250m: Global gridded soil information based on machine learning. PloS one, 12(2): e0169748.

IPCC. 2019. *2019 Refinement to the 2006 IPCC Guidelines for National Greenhouse Gas Inventories.* Switzerland, IPCC.

Jansson, C., Wullschleger, S., Kalluri, U. & Tuskan, G. 2010. Phytosequestration: Carbon Biosequestration by Plants and the Prospects of Genetic Engineering. *BioScience*, 60(9): 685–696.

Jenkinson, D. & Rayner, J. 1977. The Turnover of Soil Organic Matter in Some of The Rothamsted Classical Experiments. *Soil Science*, 123(5): 298–305.

Jenkinson D. S. 1990. The turnover of organic carbon and nitrogen in soil. *Philosophical Transactions of the Royal Society of London*, 329: 361-368.

Jenkinson, D., Adams, D. & Wild, A. 1991. Model estimates of CO2 emissions from soil in response to global warming. *Nature*, 351(6324): 304-306.

Jenkinson, D. & Coleman, K. 2008. The turnover of organic carbon in subsoils. Part 2. Modelling carbon turnover. *European Journal of Soil Science*, 59(2): 400–413.

Jenny, H., Gessel, S. & Bingham, F. 1949. Comparative Study of Decomposition Rates of Organic Matter in Temperate and Tropical Regions. *Soil Science*, 68(6): 419–432.

Jenny, H. 1994. *Factors of soil formation: A system of quantitative pedology*, 320pp. New York, USA, Dover Publications.

Keenan, T., Carbone, M., Reichstein, M. & Richardson, A. 2011. The model-data fusion pitfall: assuming certainty in an uncertain world. *Oecologia*, 167: 587-597.

Kutsch, W. L., Bahn, M. & Heinemeyer, A. 2010. *Soil carbon dynamics: An integrated methodology,* 286pp. Cambridge, UK, Cambridge University Press.

Lal, R. 2004. Soil carbon sequestration impacts on global climate change and food security. *Science*, 304(5677): 1623–1627.

Lal, R. & Follett, R. F., eds. 2009. *Soil carbon sequestration and the greenhouse effect,* 410pp.. Madison, USA, Soil Science Society of America.

Lal, R., Smith, P., Jungkunst, H., Mitsch, W., Lehmann, J., Nair, P., McBratney, A. et al. 2018. The carbon sequestration potential of terrestrial ecosystems. *Journal of Soil and Water Conservation*, 73(6): 145A–152A.

Lauenroth, W. K. and Skogerboe, G. V. & Flug, M., eds. 1983. *Analysis of ecological systems: State-of-the-art in ecological modelling,* 992pp. Amsterdam, Elsevier Scientific.

Lehtonen, A., Tupek, B., Nieminen, T., Balázs, A., Anjulo, A., Teshome, M. et al. 2020. Soil carbon stocks in Ethiopian forests and estimations of their future development under different forest use scenarios. *Land Degradation & Development*, 31(18), 2763-2774.

Li, C. 1996. The DNDC Model. In D.S. Powlson, P. SMith, J. U. Smith, eds. *Evaluation of soil organic matter models,* pp. 263-267. Berlin, Heidelberg, Springer.

Lieth, H. 1975. Modeling the primary productivity of the world. In H. Lieth & R. H. Whittaker, eds. *Primary productivity of the biosphere,* pp. 237-263. Springer, Berlin, Heidelberg.

Lieth, H. & Whittaker, R. H., eds. 1975. *Primary Productivity of the Biosphere,* 339pp. Berlin, Heidelberg, Springer.

Lorenz, K., and Lal, R., eds. 2018. *Carbon sequestration in agricultural ecosystems*, 392pp. Berlin, Heidelberg, Springer.

Lugato, E., Bampa, F., Panagos, P., Montanarella, L. & Jones, A. 2014. Potential carbon sequestration of European arable soils estimated by modelling a comprehensive set of management practices. *Global change biology*, 20: 3557-3567.

Manzoni, S. & Porporato, A. 2009. Soil carbon and nitrogen mineralization: Theory and models across scales. *Soil Biology and Biochemistry*, 41(7): 1355–1379.

Martens, R. 1995. Current methods for measuring microbial biomass C in soil: Potentials and limitations. *Biology and Fertility of Soils*, 19(2-3): 87-99.

Milne, E., Adamat, R., Batjes, N., Bernoux, M., Bhattacharyya, T., Cerri, C., Coleman, K. et al. 2007. National and sub-national assessments of soil organic carbon stocks and changes: The GEFSOC modelling system. *Agriculture, Ecosystems & Environment*, 122(1): 3–12.

Minasny, B., Malone, B., McBratney, A., Angers, D., Arrouays, D., Chambers, A., Chaplot, V. et al. 2017. Soil carbon 4 per mille. *Geoderma*, 292: 59-86.

Mondini, C., Coleman, K. & Whitmore, A. 2012. Spatially explicit modelling of changes in soil organic C in agricultural soils in Italy, 2001–2100: Potential for compost amendment. *Agriculture, Ecosystems & Environment*, 153: 24–32.

Moradizadeh, M. & Saradjian, M. 2016. Vegetation Effects Modeling in Soil Moisture Retrieval Using MSVI. *Photogrammetric Engineering & Remote Sensing*, 82(10): 803-810.

Morais, T., Teixeira, R. & Domingos, T. 2019. Detailed global modelling of soil organic carbon in cropland, grassland and forest soils. *PloS one*, 14(9): e0222604.

Motavalli, P., Palm, C., Parton, W., Elliott, E. & Frey, S. 1995. Soil pH and organic C dynamics in tropical forest soils: Evidence from laboratory and simulation studies. *Soil Biology and Biochemistry*, 27(12): 1589–1599.

Neumann, M. & Smith, P. 2018. Carbon uptake by European agricultural land is variable, and in many regions could be increased: Evidence from remote sensing, yield statistics and models of potential productivity. *Science of the total environment*, 643: 902–911.

Palosuo, T., Foereid, B., Svensson, M., Shurpali, N., Lehtonen, A., Herbst, M., Linkosalo, T. et al. 2012. A multi-model comparison of soil carbon assessment of a coniferous forest stand. *Environmental Modelling & Software*, 35: 38–49.

Parshotam, A. & Hewitt, A. 1995. Application of the Rothamsted carbon turnover model to soils in degraded semi-arid land in New Zealand. *Environment International*, 21(5): 693–697.

Parton, W., Cole, C., Stewart, J., Ojima, D. & Schimel, D. 1989. Simulating regional patterns of soil C, N, and P dynamics in the U.S. central grasslands region. In M. Clarholm & L. Bergström, eds. *Ecology of Arable Land — Perspectives and Challenges. Developments in Plant and Soil Sciences*, pp. 99-108. Doldrecht, Springer.

Parton, W. 1996. The CENTURY model. In D.S. Powlson, P. SMith, J. U. Smith, eds. *Evaluation of soil organic matter models,* pp. 283–291. Berlin, Heidelberg, Springer.

Paustian, K., Collier, S., Baldock, J., Burgess, R., Creque, J., DeLonge, M., Dungait, J. et al. 2019. Quantifying carbon for agricultural soil

management: from the current status toward a global soil information system. *Carbon Management*, 10(6): 567–587.

Paustian, K., Lehmann, J., Ogle, S., Reay, D., Robertson, G. & Smith, P. 2016. Climate-smart soils. *Nature*, 532(7597): 49–57.

Petri, M., Batello, C., Villani, R. & Nachtergaele, F. 2009. Carbon status and carbon sequestration potential in the world's grasslands. In M. Abberton, R. Conant & C. Batello, eds. *Grassland carbon sequestration: Management, policy and economics,* pp. 19-31. Rome, FAO.

Plutzar, C., Kroisleitner, C., Haberl, H., Fetzel, T., Bulgheroni, C., Beringer, T., Hostert, P. et al. 2016. Changes in the spatial patterns of human appropriation of net primary production (HANPP) in Europe 1990–2006. *Regional Environmental Change*, 16(5): 1225–1238.

Poeplau, C. & Don, A. 2012. Sensitivity of soil organic carbon stocks and fractions to different land-use changes across Europe. *Geoderma*, 192: 189–201.

Poulton, P., Johnston, J., Macdonald, A., White, R. & Powlson, D. 2018. Major limitations to achieving "4 per 1000" increases in soil organic carbon stock in temperate regions: Evidence from long-term experiments at Rothamsted Research, United Kingdom. *Global change biology*, 24(6): 2563–2584.

Powlson, D. S. and Smith, P. & Smith, J. U., eds. 1996. *Evaluation of soil organic matter models: Using existing long-term datasets,* 433pp. Berlin, Heidelberg, Springer.

Richter, J. 1981. Simulation of nitrogen behaviour of soil-plant systems: Papers of a workshop models for the behaviour of nitrogen in soil and uptake by plant. *Zeitschrift f{ür Pflanzenern{ährung und Bodenkunde*, 144: 428-429.

Riggers, C., Poeplau, C., Don, A., Bamminger, C., Höper, H. & Dechow, R. 2019. Multi-model ensemble improved the prediction of trends in soil organic carbon stocks in German croplands. *Geoderma*, 345: 17-30.

Saggar, S., Parshotam, A., Sparling, G., Feltham, C. & Hart, P. 1996. 14C-labelled ryegrass turnover and residence times in soils varying in clay content and mineralogy. *Soil Biology and Biochemistry*, 28(12): 1677–1686.

Scharlemann, J., Tanner, E., Hiederer, R. & Kapos, V. 2014. Global soil carbon: understanding and managing the largest terrestrial carbon pool. *Carbon Management*, 5(1): 81-91.

Schmer, M., Jin, V., Wienhold, B., Varvel, G. & Follett, R. 2014. Tillage and Residue Management Effects on Soil Carbon and Nitrogen Under Irrigated Continuous Corn. *Soil Science Society of America Journal*, 78: 1987-1996.

Schmidt, M., Torn, M., Abiven, S., Dittmar, T., Guggenberger, G., Janssens, I., Kleber, M. et al. 2011. Persistence of soil organic matter as an ecosystem property. *Nature*, 478(7367): 49–56.

Schulze, E., Ciais, P., Luyssaert, S., Schrumpf, M., Janssens, I., Thiruchittampalam, B., Theloke, J. et al. 2010. The European carbon balance. Part 4: integration of carbon and other trace-gas fluxes. *Global change biology*, 16(5): 1451-1469.

Shang, C. & Tiessen, H. 1998. Organic Matter Stabilization in Two Semiarid Tropical Soils: Size, Density, and Magnetic Separations. *Soil Science Society of America Journal*, 62(5): 1247–1257.

Shirato, Y., Hakamata, T. & Taniyama, I. 2004. Modified rothamsted carbon model for andosols and its validation: changing humus decomposition rate constant with pyrophosphate-extractable Al. *Soil Science and Plant Nutrition*, 50(1): 149–158.

Shirato, Y. & Yokozawa, M. 2005. Applying the Rothamsted Carbon Model for long-term experiments on Japanese paddy soils and modifying it by simple tuning of the decomposition rate. *Soil Science and Plant Nutrition*, 51: 405-415.

Sierra, C., Müller, M. & Trumbore, S. 2012. Models of soil organic matter decomposition: the SoilR package, version 1.0. *Geoscientific Model Development*, 5(4): 1045–1060.

Sinclair, T. & Seligman, N. 1996. Crop Modeling: From Infancy to Maturity. *Agronomy Journal*, 88(5): 698–704.

Six, J., Conant, R., Paul, E. & Paustian, K. 2002. Stabilization Mechanisms of Soil Organic Matter: Implications for C-Saturation of Soils. *Plant and Soil*, 241(2): 155–176.

Smith, P., Smith, J., Powlson, D., McGill, W., Arah, J., Chertov, O., Coleman, K. et al. 1997. A comparison of the performance of nine soil organic matter models using datasets from seven long-term experiments. *Geoderma*, 81(1-2): 153-225.

Smith, P., Andrén, O., Brussaard, L., Dangerfield, M., Ekschmitt, K., Lavelle, P. & Tate, K. 1998. Soil biota and global change at the ecosystem level: describing soil biota in mathematical models. *Global Change Biology*, 4(7): 773–784.

Smith, P. & Falloon, P. 2000. Modelling refractory soil organic matter. *Biology and Fertility of Soils*, 30(5-6): 388-398.

Smith, P. 2004. How long before a change in soil organic carbon can be detected?. **Global Change Biology**, 10(11): 1878-1883.

Smith, J., Smith, P., Wattenbach, M., Zaehle, S., Hiederer, R., Jones, R., Montanarella, L. et al. 2005. Projected changes in mineral soil carbon of European croplands and grasslands, 1990-2080. *Global Change Biology*, 11(12): 2141–2152.

Smith, P., Smith, J., Franko, U., Kuka, K., Romanenkov, V., Shevtsova, L., Wattenbach, M. et al. 2007. Changes in mineral soil organic carbon stocks in the croplands of European Russia and the Ukraine, 1990–2070; comparison of three models and implications for climate mitigation. *Regional Environmental Change*, 7(2): 105–119.

Smith, J., Smith, P., Jeannette Meyer, M., Zaehle, S., Robert J.A. Jones, R., Mark Rounsevell, L. et al. 2006. Projected changes in mineral soil carbon of European forests, 1990-2100. *Canadian Journal of Soil Science*, 86: 159-169.

Smith, P., Soussana, J.F., Angers, D., Schipper, L., Chenu, C., Rasse, D., Batjes, N. et al. 2020. How to measure, report and verify soil carbon change to realize the potential of soil carbon sequestration for atmospheric greenhouse gas removal. *Global Change Biology*, 26(1): 219–241.

Smith, P., Martino, D., Cai, Z., Gwary, D., Janzen, H., Kumar, P., McCarl, B. et al. 2008. Greenhouse gas mitigation in agriculture. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 363(1492): 789–813.

Tifafi, M., Guenet, B. & Hatté, C. 2018. Large Differences in Global and Regional Total Soil Carbon Stock Estimates Based on SoilGrids, HWSD, and NCSCD: Intercomparison and Evaluation Based on Field Data From USA, England, Wales, and France. *Global Biogeochemical Cycles*, 32(1): 42–56.

Vries, W. 2018. Soil carbon 4 per mille: a good initiative but let's manage not only the soil but also the expectations. *Geoderma*, 309: 111–112.

Weihermüller, L., Graf, A., Herbst, M. & Vereecken, H. 2013. Simple pedotransfer functions to initialize reactive carbon pools of the RothC model. *European Journal of Soil Science*, 64(5): 567–575.

Wieder, W., Grandy, A., Kallenbach, C. & Bonan, G. 2014. Integrating microbial physiology and physio-chemical principles in soils with the MIcrobial-MIneral Carbon Stabilization (MIMICS) model. *Biogeosciences*, 11(14): 3899–3917.

Wiesmeier, M., Schad, P., Lützow, M., Poeplau, C., Spörlein, P., Geuß, U., Hangen, E. et al. 2014. Quantification of functional soil organic carbon pools for major soil units and land uses in southeast Germany (Bavaria). *Agriculture, Ecosystems & Environment*, 185: 208-220.

Wiesmeier, M., Poeplau, C., Sierra, C., Maier, H., Frühauf, C., Hübner, R., Kühnel, A. et al. 2016. Projected loss of soil organic carbon in temperate agricultural soils in the 21(st) century: effects of climate change and carbon input trends. *Scientific reports*, 6: 32525.

Williams, J., Dyke, P. & Jones, C. 1983. Epic - a Model for Assessing the Effects of Erosion on Soil Productivity. *Journal of Soil and Water Conservation*, 38(5): 381-383.

Yigini, Y., Olmedo, G., Reiter, S., Baritz, R., Viatkin, K., and Vargas, R. 2018. *Soil organic carbon mapping cookbook*, 223pp. Rome, FAO. (available at https://www.fao.org/3/I8895EN/i8895en.pdf)

Annex I: troubleshooting

Common errors and considerations

The following section provides an overview of common issues and potential solutions.

1. SOC stocks at T_0, Final SOC stocks, and Sequestration Rates with significantly lower values than expected (refer to section 16.1)

→ *Possible reason & solution:* The SOC and Clay layer were in the wrong unit (e.g. the input SOC layer was in kg/m^2 instead of t/ha and Clay was in g/kg instead of percent). The units need to be corrected and the whole process has to be repeated.

2. High percent of -999 values, negative/positive values outside the expected ranges that do not show a clear dispersion pattern

→ *Possible reason & solution:* The input data was missing for several pixels, and/or the differential equations calculated with the "euler" method did not generate results. If these values do not exceed 10 percent of all pixel values and are dispersed without any clear pattern they can be simply masked out.

3. More than 10 percent of clustered *No data values* (values equal to -999) and negative/positive values outside the expected ranges

→ *Possible reason & solution:* Potential errors in the input data e.g. missing values and/or wrong units are present. Additionally, errors can occur due to potential model limitations including:

- Model limitations when simulating changes in soils with high initial SOC content (>200 t/ha); percent clay).
- Model limitations when simulating changes in tropical conditions.

If more than 10 percent of the modeled area shows no values, the procedure should be re–run using the "lsoda" method in the SoilR package to solve the differential equations in scripts 13 to 15. For more information on the limitations of the RothC model please refer to chapter 14.

4. Very high uncertainty values → *Possible reason & solution:* Minimum and maximum values used for the various input layers (e.g. GSOCmap, Clay, Temperature and Precipitation) when running the model should be checked. They should represent the upper and lower limits of the 95 percent confidence interval. For more information on the approach used for the estimation of uncertainty please refer to chapter 12.

Annex II: quality assurance and quality control

The following protocol was devised to provide National Experts with a step–by–step guideline to perform a Quality Assurance (QA) and Quality Control (QC) of the 29 GSOCseq products that result from the proposed methodology.

The following protocol does not provide any guidance in terms of uncertainty estimation and validation. For more details and information on the estimation of uncertainties and potential map validation strategies please refer to Chapter 12.

Quality assurance and quality control consist of activities to ensure the quality of a particular result. Quality control is a reactive process that focuses on identifying defects and errors while quality assurance is a proactive approach aimed at preventing defects and errors. In the context of digital soil mapping, both processes are often interlinked. A QA is interlinked with a QC when it identifies defects and the QA remodels the process to eliminate the defects and prevent them from recurring (Chapman, 2005)(Figure 16.1).

Each step in the following protocol should be considered in order to detect and eliminate errors, address data inaccuracies and assess the output completeness.

Quality Assurance & Quality Control

Step 1 Check that the 29 products have been labeled appropriately and are in the correct format

The following Table (Table 16.1) gives an overview of all the GSOCseq products in alphabetical order. Each product should include the ISO 3166-1 alpha-3 country code in its name. For instance, in the case of Turkey, ISO_GSOCseq_AbsDiff_BAU_Map030 should be changed to TUR_GSOCseq_AbsDiff_BAU_Map030.

All 29 products must be georeferenced TIF (.tif) files.

Overview of the 29 GSOCseq Products.

Product.overview	Formula
a. Absolute differences in SOC stocks: ISO GSOCseq_AbsDiff_BAU_Map030 ISO GSOCseq_AbsDiff_SSM1_Map030 ISO GSOCseq_AbsDiff_SSM2_Map030 ISO GSOCseq_AbsDiff_SSM3_Map030	Scenario - Carbon at T0
b. Absolute sequestration rates with uncertainties: ISO GSOCseq_ASR_BAU_Map030 ISO GSOCseq_ASR_BAU_UncertaintyMap030 ISO GSOCseq_ASR_SSM1_Map030 ISO GSOCseq_ASR_SSM1_UncertaintyMap030	(Scenario - Carbon at T0)/20
ISO GSOCseq_ASR_SSM2_Map030 ISO GSOCseq_ASR_SSM2_UncertaintyMap030 ISO GSOCseq_ASR_SSM3_Map030 ISO GSOCseq_ASR_SSM3_UncertaintyMap030 c. Final SOC stocks after 20 years with uncertainties:	
ISO GSOCseq_BAU_UncertaintyMap030 ISO GSOCseq_SSM_UncertaintyMap030 ISO GSOCseq_finalSOC_BAU_Map030 ISO GSOCseq_finalSOC_SSM1_Map030 ISO GSOCseq_finalSOC_SSM2_Map030	
ISO GSOCseq_finalSOC_SSM3_Map030 d. Relative differences: ISO GSOCseq_RelDiff_SSM1_Map030 ISO GSOCseq_RelDiff_SSM2_Map030 ISO GSOCseq_RelDiff_SSM3_Map030	BAU - SSM scenario
e. Relative sequestration rates with uncertainties: ISO GSOCseq_RSR_SSM1_Map030 ISO GSOCseq_RSR_SSM1_UncertaintyMap030 ISO GSOCseq_RSR_SSM2_Map030 ISO GSOCseq_RSR_SSM2_UncertaintyMap030	(BAU - SSM scenario)/20
ISO GSOCseq_RSR_SSM3_Map030 ISO GSOCseq_RSR_SSM3_UncertaintyMap030 f. Initial SOC stocks at year 2020 (T0) with uncertainties: ISO_GSOCseq_T0_Map030 ISO_GSOCseq_T0_UncertaintyMap030	

Step 2 Check the projection and resolution of all products

Open the products in QGIS or any other preferred GIS platform. Check that the projection of all products is EPSG:4326 – WGS 84 (Layer properties). Check that the spatial resolution (pixel size) (Layer properties) is equal to ~0.00833 degrees ; 1 km x 1km at the equator.

Step 3 Check that the products were generated for agricultural and grazing areas only

Visualize the 29 products in QGIS or any preferred GIS platform. Load a land–use layer to visually assess that the simulations were done exclusively on agricultural areas.

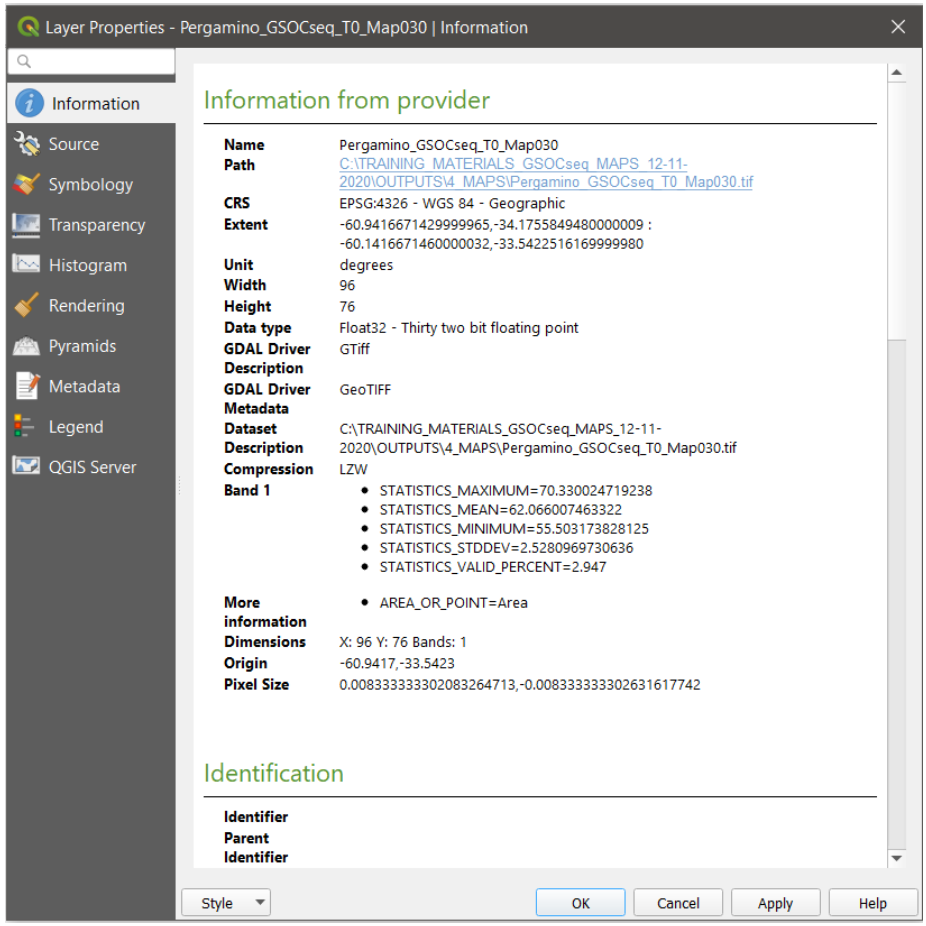
Step 4 Check for units, range, and outliers

In the following section expected value ranges for each product category are presented. It is important to note that the provided ranges represent a gross approximation of what can be normally expected when running the proposed methodology. Results that fall outside these ranges and that did not occur due to common issues presented in the section 15.2 need to be carefully evaluated based on local expertise and available literature.

The provided ranges can be compared in QGIS, R, or any preferred platform. Descriptive layer statistics can be viewed in QGIS under Layer Properties in the Information tab (Figure 16.2).

Soil organic carbon at T0 (2020)

The product *ISO_GSOCseq_T0_Map030* refers to the SOC stocks 0–30 cm in t/ha used as input in the Forward phase.



Layer statistics in QGIS

Check statistics of ISO_GSOCseq_T0_Map030 file. The provided reference values may vary for each country depending on external factors that need to be carefully evaluated. However, the following values can be generally expected:

- Most of the values should fall between 15–100 t/ha
- Minimum values should be greater than 0 (except for -999 Values, which indicate no data values)
- -999 values should be masked out
- There should not be negative values other than -999
- Maximum values should not exceed 800 t/ha.

Final SOC stocks after 20 simulation years for all scenarios (BAU, SSM1, SSM2, SSM3)

These products refer to the SOC stocks 0–30 cm in t/ha at the end of the forward phase simulations for the 4 scenarios (BAU, SSM1, SSM2, SSM3).

Check the statistics of:

1. ISO_GSOCseq_finalSOC_BAU_Map030
2. ISO_GSOCseq_finalSOC_SSM1_Map030
3. ISO_GSOCseq_finalSOC_SSM2_Map030
4. ISO_GSOCseq_finalSOC_SSM3_Map030

The provided reference values may vary for each country depending on external factors that need to be carefully evaluated. However, the following values can be generally expected:

- Most of the values should fall between 15–100 t/ha
- Minimum values should be greater than 0 (except for -999 Values, which indicate no data values)
- -999 values should be masked out
- There should not be negative values other than -999
- Maximum values should not exceed 800 t/ha.
- Mean values SOC SSM3 > SSM 2 > SSM 1 > BAU

Absolute differences in SOC stocks

The layers showing the absolute differences in SOC stocks refer to the SOC change 0–30 cm in t/ha (Final SOC stocks vs initial stocks T0) for the BAU, SSM1, SSM2, and SSM3 scenarios.

Check the statistics of:

1. ISO_GSOCseq_AbsDiff_BAU_Map030
2. ISO_GSOCseq_AbsDiff_SSM1_Map030
3. ISO_GSOCseq_AbsDiff_SSM2_Map030
4. ISO_GSOCseq_AbsDiff_SSM3_Map030

The provided reference values may vary for each country depending on external factors that need to be carefully evaluated. However, the following values can be generally expected:

- The expected range should fall usually between -80 to +80 t/ha (with an approximate average of -15 to +15)
- Negative values can occur and indicate SOC losses between 2020 and 2040 (-999 indicate no data values).
- -999 values should be masked out
- Negative values other than -999 values should usually not exceed -80
- Maximum values should usually not exceed +80
- Check if extreme values (values > +80 and values < -80) are grouped or dispersed (refer to Annex I, Common errors and considerations)
- Mean values in order of size SSM3 > SSM 2 > SSM 1 > BAU

Relative differences in SOC stocks

These products refer to the SOC change 0–30 cm in t/ha (Final SOC stocks in SSM scenarios vs Final stocks BAU scenario) for the SSM1, SSM2, and SSM3 scenarios.

Check the statistics of:

1. ISO_GSOCseq_RelDiff_SSM1_Map030
2. ISO_GSOCseq_RelDiff_SSM2_Map030

3. ISO_GSOCseq_RelDiff_SSM3_Map030

The provided reference values may vary for each country depending on external factors that need to be carefully evaluated. However, the following values can be generally expected:

- Expected range usually falls between 0 to +80 t/ha
- -999 values should be masked out
- There should not be negative values other than -999
- Maximum values should usually not exceed +80
- Check if extreme values (values > +80) are grouped or dispersed (refer to Annex I, Common errors and considerations)
- Mean values SSM3 > SSM 2 > SSM 1

Absolute sequestration rates (ASR)

Absolute sequestration rates (ASR) refer to the SOC change rate 0–30 cm in t/ha/yr (Final SOC stocks vs initial stocks T0 divided by 20 years) for the BAU, SSM1, SSM2, and SSM3 scenarios.

Check the statistics of:

1. ISO_GSOCseq_ASR_BAU_Map030
2. ISO_GSOCseq_ASR_SSM1_Map030
3. ISO_GSOCseq_ASR_SSM2_Map030
4. ISO_GSOCseq_ASR_SSM3_Map030

The provided reference values may vary for each country depending on external factors that need to be carefully evaluated. However, the following values can be generally expected:

- The expected range for all maps should fall between -4 to +4 t/ha
- ASR BAU: usually most values from -0.5 to + 0.5, with median values near 0 or lower
- ASR SSM1: usually most values -0.4 to + 0.6, with median near 0 or higher (similar to BAU)
- ASR SSM2: usually most values -0.3 to + 0.7

- ASR SSM3: usually most values -0.2 to + 0.8
- -999 and -49.95 Values (-999/20) indicate no data values. Values <= -49.95 should be masked out
- Negative values other than -999 and -49.95 (meaning SOC losses between 2020 and 2040) should not exceed -4
- Maximum values should usually not exceed +4.
- Check if extreme values (> +4; <-4) are grouped or dispersed (refer to Annex I, Common errors and considerations)
- Mean values SSM3 > SSM 2 > SSM 1 > BAU

Relative sequestration rates (RSR)

Relative sequestration rates refer to the SOC change rate 0–30 cm in t/ha/yr (Final SOC stocks under SSM vs Final stocks BAU divided by 20 years) for the SSM1, SSM2, and SSM3 scenarios, compared to BAU scenario.

Check the statistics of:

1. ISO_GSOCseq_RSR_SSM1_Map030
2. ISO_GSOCseq_RSR_SSM2_Map030
3. ISO_GSOCseq_RSR_SSM3_Map030

The provided reference values may vary for each country depending on external factors that need to be carefully evaluated. However, the following values can be generally expected:

- The expected range should fall between 0 to +4 t C/ha (with most data being distributed between 0 to +1)
- RSR SSM1: usually most values range from 0 to + 0.6, with median near 0 or higher (similar to BAU)
- RSR SSM2: usually most values range from 0 to + 0.7
- RSR SSM3: usually most values range 0 to + 0.8
- -999 and -49.95 Values (-999/20) indicate no data values. Values <= -49.95 will be excluded from Global product
- There should not be negative values other than -999 and -49.95:
- Check if grouped or dispersed (refer to Annex I)
- Maximum values should usually not exceed +4.
- Check if grouped or dispersed (refer to Annex I)
- Mean SOC values in order of size SSM3 > SSM 2 > SSM 1

Uncertainties

Check the statistics of:

1. ISO_GSOCseq_ASR_BAU_UncertaintyMap030
2. ISO_GSOCseq_ASR_SSM1_UncertaintyMap030
3. ISO_GSOCseq_ASR_SSM2_UncertaintyMap030
4. ISO_GSOCseq_ASR_SSM3_UncertaintyMap030
5. ISO_GSOCseq_BAU_UncertaintyMap030
6. ISO_GSOCseq_RSR_SSM1_UncertaintyMap030
7. ISO_GSOCseq_RSR_SSM2_UncertaintyMap030
8. ISO_GSOCseq_RSR_SSM3_UncertaintyMap030
9. ISO_GSOCseq_SSM_UncertaintyMap030
10. ISO_GSOCseq_T0_UncertaintyMap030

The provided reference values may vary for each country depending on external factors that need to be carefully evaluated. However, the following values can be generally expected:

- Expected range all maps, usually between 0 to +200 percent (most data generally between 15 to 50 percent)
- -999 indicates no data values. Negative values and -999 values should be masked out
- Maximum values should usually not exceed +200 percent.
- Check if extreme values (>200 percent and <0 percent) are grouped or dispersed (refer to Annex I, Common errors and considerations)

The Global Soil Partnership (GSP) is a globally recognized mechanism established in 2012. Our mission is to position soils in the Global Agenda through collective action. Our key objectives are to promote Sustainable Soil Management (SSM) and improve soil governance to guarantee healthy and productive soils, and support the provision of essential ecosystem services towards food security and improved nutrition, climate change adaptation and mitigation, and sustainable development.

Thanks to the financial support of

European Commission

Ministry of Finance of the Russian Federation

Rural Development Administration

ISBN 978-92-5-133796-7

CB2642EN/1/01.22