Politecnico di Milano

Course of Civil Engineering

A.Y. 2014-2015

Reducing water footprint: more crop per drop

Relatore: Prof.ssa Maria Cristina RULLI

Correlatori: Kyle DAVIS, Paolo D'ODORICO

Antonio Giovanni Seveso

Matr. n. 783458

Abstract

Evaluate water need for a crop in a quick and efficient way it's a necessity nowadays even more requested in hydrology. Being able to do these evaluations quickly, allows us to simulate and understand different kind of scenarios, actual and futuristic, in a very efficacious way.

The aim of this study is, indeed, to create a tool, in a shape of a website, easy to use, that gives real and true values, but at the same time it has to be fully customizable from the user, both in his actual version and in possible future improvements. The study is then concluded with the analysis of actual world situation in terms of agricultural production (looking at 14 most cultivated crops), to prove that, redistributing in an efficient way those crops between the countries, it's possible to increase the production in terms of kilograms, kilocalories and proteins, reducing the utilize of water and without adding a single hectare of land to the current cultivated land in the world.

At the end of this work, there are also presented some possible scenarios of future improving.

Stimare la necessità di acqua per una coltivazione, in maniera rapida ed accurata, è una necessità sempre più richiesta in ambito idrologico. Riuscire a far ciò in tempi brevi permette di simulare e comprendere scenari attuali e futuri in maniera rapida ed efficace.

Lo scopo di questo studio è, appunto, quello di creare un tool, nella forma di un sito web, facile da usare, che fornisca dati reali e credibili, e allo stesso tempo interamente personalizzabile da parte dell'utente, sia nella sua versione attuale, sia in possibili sviluppi futuri. Lo studio viene poi concluso con l'analisi dell'attuale situazione mondiale in termini di produzione agricola (analizzando le 14 coltivazioni più diffuse), per dimostrare che, ridistribuendo in maniera efficace queste coltivazioni nel mondo, è possibile incrementare la produzione in termini di chilogrammi, chilocalorie e proteine, riducendo l'utilizzo di acqua e senza aumentare di un solo ettaro la superficie della terra attualmente coltivata.

A conclusione di questo studio, vengono presentati possibili scenari di sviluppo del progetto, migliorabile in tantissimi modi differenti.

Index

1. Introduction

Evaluating water need for a crop it's not so easy, especially if we want to simulate it with a computer, rather than making a real experiment. The benefits of a computer simulation are a lot:

- Less time consumption: a few seconds rather than a few weeks or months;
- Possibility to run different simulations in different climatic regions;
- Evaluate climate changing effects.

Of course, setting up a simulation requests a scientific approach that will end into a physical model. This model, translated into a programming language, will allow us to run the simulation. We have to make sure that:

- The results of the simulations have to be as close as possible to values of real life, and if they're not the same, we need to give a percentage of variation that we'll define "acceptable";
- The simulations must run under certain conditions and situations, that we have to define clearly and without interpretations.

In this study, our goal is to create a tool, with the layout of a website tool, that will allow everyone to calculate easily crop's need, under the assumptions we'll present in the next pages.

The tool, publicly accessible through the website www.watergrab.zone, it's a recreation of different FAO tools, such as *CropWat* and *ClimWat*, into a single software. To reach this goal, we've used the Penman-Monteith equation, well described in the next chapter, with some approximations, and a lot of different datasets, each one for a specific need:

WaterGrab

Online tool for evaluating crops' irrigation and evapotranspiration needings

- **climate datas**, such as windspeed, precipitation, temperature, etc
- **Soil datas**, that is the composition of the soil in each part of the world
- **Crop datas**, that are the coefficients we need to evaluate the evapotranspiration of the crop in the assumed climatological and soil conditions

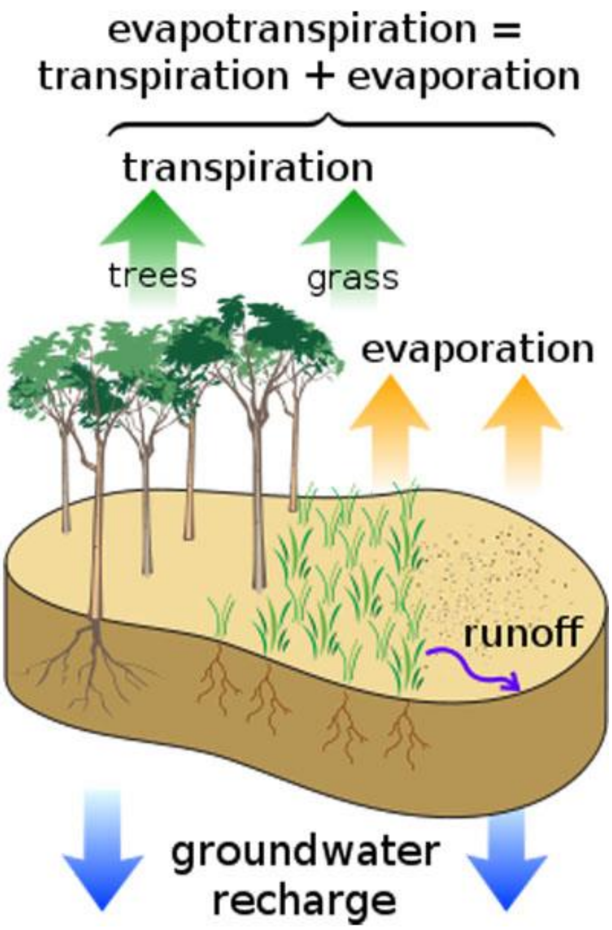
Once we've set up our tool, we then focus our attention to answer another question of utilizing of water: can we reduce the water consumption, keeping (or improving) the current production? Basically, we based our future studies on 4 different values:

- the amount of food produced [*ton*];
- the amount of kilocalories produced [*kcal*];
- the amout of grams of proteins produced [g_{prot}];
- the amout of water used [m^3].

Taking under consideration all these factors, with other assumptions, we've found a theoretical distribution of crops that will maximize the production, reducing the water consumption.

To connect the two precedent steps (watergrab.zone website and minimizing of water footprint), we'll finish this study calculating the water need of some current cultivated crops in 15 countries that we've chosen, and see what's the water need to close the yield gap.

2. Penman-Monteith equation



With this first part we want to present the differences between what we'll call ET_0, that is the reference crop evapotranspiration, and ET_C, that is the crop evapotranspiration under standard conditions.

We'll start presenting the evapotranspiration process, from its physical point of view, to the see the mathematical approach of Penman-Monteith equations.

Evapotranspiration Process

The evapotranspiration process, as it's said by the word, it's the combination of two processes:

- evaporation, that's the process whereby water is lost from the soil surface
- transpiration, that's the process that lose water through the crops.

To change the state of molecules of water, we need energy, given by the solar radiation and, of course, by the ambient temperature. While the water evaporates, the air surrounding becomes more and more saturated, slowing down the evaporation process, till a point where the evaporation is stopped for saturation of air (100% humidity). Wind speed plays an important role, in fact it guarantees the replacement of saturated air with unsaturated air.

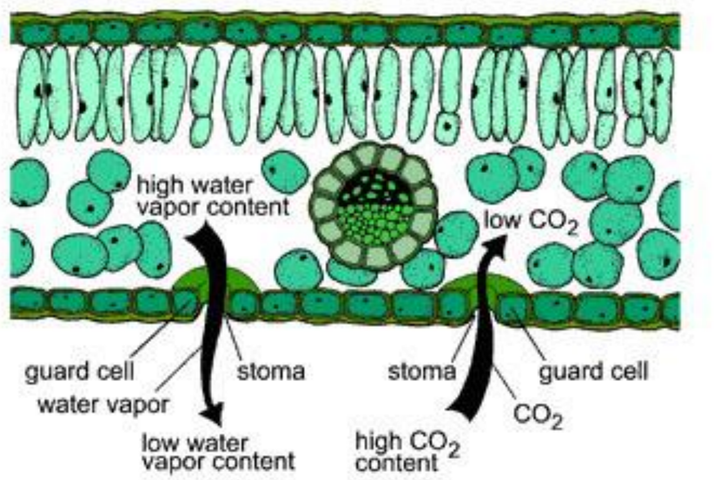
This first short consideration makes us aware about some of the main factors that we'll take under consideration during our calculations:

- solar radiation;

- air temperature;
- air humidity;
- wind speed.

The transpiration is basically the vaporization of liquid water contained in plant tissues. Crops mainly lose their water through stomata, that are very small openings on the plant leaf through which gases and water vapor pass.

Inside a crop, water is taken up to leaves from root through the plant, bringing also nutrients. So the transpiration is controlled by leaves or, to be more precise, by stomata openings. All the water taken up by the crop is lost by transpiration and just a little fraction is used within the crop.



Of course, as well as for the evaporation, transpiration depends on energy supply, humidity of the air and wind speed. Thus, we can consider the same factor seen above for the evaporation.

Another factor, that we cannot omit, is the amount of water available. If water is too deep inside the soil, we cannot consider evaporation, as well as crop is not able to get the water needed to survive. So, we can also add the factor of soil water content and the ability of the soil to conduct water to the roost.

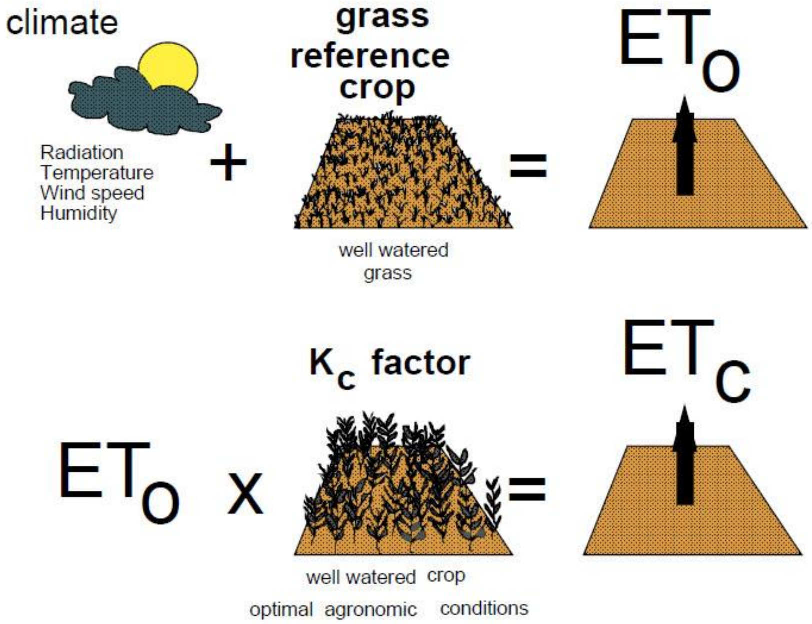
Transpiration, finally, is not the same for all the crops. Different kinds of plants may have different transpiration rates.

Evapotranspiration is the combination of the above described factors. There is no way to distinguish between the two processes. At the initial stage of crop, the water is mainly lost by evaporation. With the crop growing, it covers completely the soil, reducing the solar radiation reaching the ground. That will consist in more transpiration and less evaporation. The evapotranspiration rate is normally expressed in millimeters $[mm]$ per unit time.

When we talk about ET_0, we express the evaporation power of atmosphere. ET_C refers to the evapotranspiration from excellently managed, large, well-watered fields that achieve full production under the given climatic conditions.

The **reference crop evapotranspiration** ET_0 has a reference surface that is a hypothetical grass reference crop with specific characteristics. This concept was introduced to study the evaporative demand of the atmosphere independently of crop type, crop development and management practices.

The only factors affecting ET_0 are climatic parameters. Consequently, ET_0 is a climatic parameter and can be computed from weather data. ET_0 express the evaporating power of the atmosphere at a specific location and time of the year and does not consider the crop characteristics and soil factors. To evaluate this parameter, we'll use the FAO Penman-Monteith method, that has been selected because it closely approximates grass ET_0 at the location evaluated, is physically based, and explicitly incorporates both physiological and aerodynamic parameters. Moreover, procedures have been developer for estimating missing climatic parameters.



The **crop evapotranspiration under standard conditions**, well known as ET_C, is the evapotranspiration from disease-free, well-fertilized crops, grown in large fields, under optimum soil water conditions, and achieving full production under the given climatic conditions.

The amount of water required to compensate the evapotranspiration loss from the cropped field is defined as crop water requirement. The irrigation water requirement basically represents the difference between the crop water requirement and effective precipitation. The irrigation water requirement also includes additional water for leaching of salts and to compensate for non-uniformity of water application.

ET_C can be calculated from climatic data and by integrating directly the crop resistance in the Penman-Monteith approach. We can easily sum up the approach in the following equation:

$$ET_C = ET_0 \cdot K_C$$

K_C is not constant from sowing till harvest, but it varies in different periods of growing season. It varies also from crop to crop due to differences in leaf anatomy, stomatal characteristics and aerodynamic properties.

If we want to be even more precise, we can then consider **crop evapotranspiration under non-standard conditions** ($ET_{C\,adj}$). This is the evapotranspiration from crops grown under management and environmental conditions that differ from the standard conditions. For example, the presence of pests and diseases, soil salinity or low soil fertility will determine a variation of evapotranspiration. Following the precedent approach, we can evaluate $ET_{C\,adj}$ with the following equation

$$ET_C = ET_0 \cdot K_C \cdot K_S$$

Theoretical equations

Different methods have been presented over the last 60 years. The one that was considered to offer the best results with minimum possible errors in relation to a living grass reference crop, was the Penman-Monteith one.

In 1948 Penman combined the energy balance with the mass transfer method and derived an equation to compute the evaporation from an open water surface from standard climatological records of sunshine, temperature, humidity and wind speed. This so-called combination method was further developed by many researchers and extended to cropped surfaces by introducing resistance factors.

The basic equation for Penman-Monteith we start looking at is:

$$\lambda ET = \frac{\Delta \cdot (R_n - G) + \rho_a c_p \frac{(e_s - e_a)}{r_a}}{\Delta + \gamma\left(1 + \frac{r_s}{r_a}\right)}$$

Where the values are the following:

- R_n is the net radiation
- G is the soil heat flux
- $(e_s - e_a)$ represents the vapor pressure deficit of the air
- ρ_a is the mean air density at constant pressure
- c_p is the specific heat of the air
- Δ represents the slope of the saturation vapor pressure temperature relationship
- γ is the psychrometric constant
- r_s is tge (bulk) surface
- r_a is the aerodynamic resistance

As we can see, this approach ass formulated above, includes all parameters that govern energy exchange and corresponding latent heat flux (evapotranspiration) from uniform expanses of vegetation.

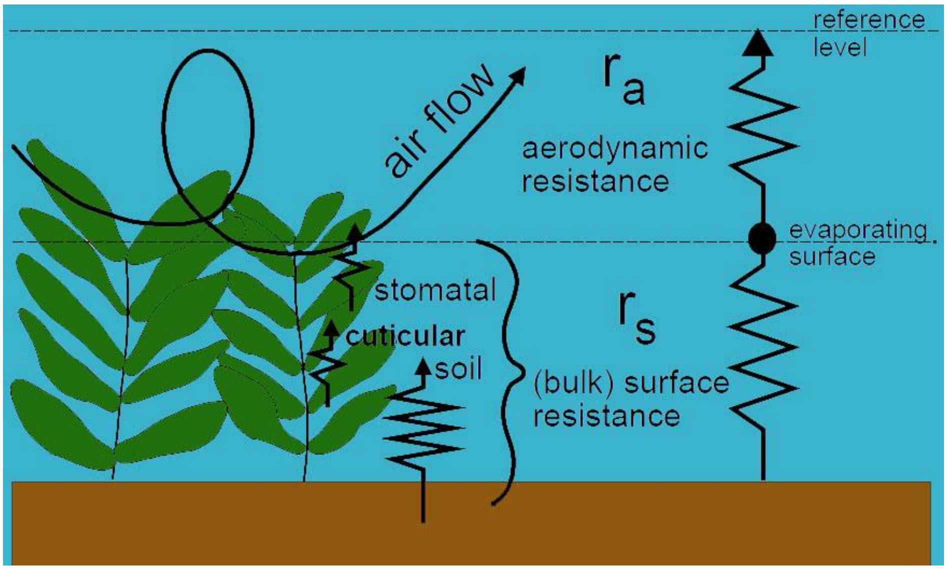
To give a more detailed look to all the above parameters and to be able to evaluate them, we'll discuss the step by step as follows:

The **aerodynamic resistance (r_a)** determine the transfer of heat and water vapor from the evaporating surface into the air above the canopy. The formulation is the following:

$$r_a = \frac{\ln\left[\frac{z_m - d}{z_{om}}\right]\ln\left[\frac{z_h - d}{z_{oh}}\right]}{k^2 u_z}$$

Where:

- r_a is the aerodynamic resistance $\left[\frac{s}{m}\right]$
- z_m is the height of wind measurements $[m]$
- z_h is the height of humidity measurements $[m]$
- d is the zero plane displacement height $[m]$
- z_{om} is the roughness length governing momentum transfer $[m]$
- z_{oh} is the roughness length governing transfer of heat and vapor $[m]$
- k is the Von Karman's constant, equal to 0.4 $[-]$
- u_z is the wind speed at height z $\left[\frac{m}{s}\right]$



The equation is restricted for neutral stability conditions, that is where temperature, atmospheric pressure and wind velocity follow nearly adiabatic conditions (no heat exchange). If we want to use this equation for shot time periods (hours or less), we need to include corrections for stability. However, for ET_0 the heat exchanged is normally small, and therefore stability correction is normally not required.

The **(bulk) surface resistance (r_s)** describes the resistance of vapor flow through the transpiring crop and evaporating soil surface. Where the vegetation does not completely cover the soil, the resistance factor should indeed include the effects of the evaporation from the soil surface. If the crop is not transpiring at a potential rate, the resistance depends also on the water status of the vegetation. A good approximation of this factor can be the following equation:

$$r_s = \frac{r_l}{LAI_{active}}$$

Where:

- r_s (bulk) surface resistance $\left[\frac{s}{m}\right]$
- r_l bulk stomatal resistance of the well-illuminated leaf $\left[\frac{s}{m}\right]$
- LAI_{active} active (sunlit) leaf area index $\left[\frac{m^2_{leaf\ area}}{m^2_{soil\ surface}}\right]$

The *Leaf Area Index* is the leaf area (upper side only) per unit area of soil below. Values of $3 \div 5$ are common for many mature crops.

Also the *bulk stomatal resistance* r_l differs from crop to crop. It usually increases as the crop ages. It is influenced by climate and by water availability, but the rate of influence vary from one crop to another.

As we have done till there, we are looking to a formulation where we have ET and not ET_0. Relating ET_0 to a specific crop has the advantage of incorporating the biological and physical processes involved in ET from cropped surfaces.

Grass, together with alfalfa, is a well-studied crop regarding its aerodynamic and surface characteristics, being nowadays accepted worldwide as a reference surface. This is perfect for the characteristics of the reference crop, that have to be well defined and fixed. This mainly because the resistance to diffusion of vapor strongly depends on crop height, ground cover, *LAI* and soil moisture conditions. This led to select a hypothetical grass reference, with an assumed crop height of $0.12\ [m]$, a fixed surface resistance of $\left[\frac{s}{m}\right]$ and an albedo of $0.23''$.

This reflects an extensive surface of green grass of uniform height, actively growing, completely shading the ground and with adequate water.

Actual Penman-Monteith equation

In 1990 a consultation of experts and researchers recommended the adoption of the Penman-Monteith combination methods as a new standard for reference evapotranspiration and advised on procedures for calculation of the various parameters.

From the original Penman-Monteith equation reported above, and the equations of the aerodynamic (r_a) and surface (r_s) resistance, we got the FAO Penman-

Monteith method to estimate ET_0 as follows

$$ET_0 = \frac{0.408\Delta(R_n - G) + \gamma \frac{900}{T + 273} u_2 (e_s - e_a)}{\Delta + \gamma(1 + 0.34u_2)}$$

Where:

- ET_0 reference evapotranspiration $\left[\frac{mm}{day}\right]$
- R_n net radiation at the crop surface $\left[\frac{MJ}{m^2 day}\right]$
- G soil heat flux density $\left[\frac{MJ}{m^2 day}\right]$
- T mean daily air temperature at $2\ [m]$ height $[°C]$
- u_2 wind speed at $2[m]$ height $\left[\frac{m}{s}\right]$
- e_s saturation vapor pressure $[kPa]$
- e_a actual vapor pressure $[kPa]$
- $e_s - e_a$ saturation vapor pressure deficit $[kPa]$
- Δ slope vapor pressure curve $\left[\frac{kPa}{°C}\right]$
- γ psychrometric constant $\left[\frac{kPa}{°C}\right]$

This approach to evapotranspiration with ET_0 allows us to compare evapotranspiration at different periods of the year on in different regions, as well as we can relate evapotranspiration of other crops.

In the equation we'll use standard climatological records of solar radiation (sunshine), air temperature, humidity and wind speed. To ensure the integrity of computations, the weather measurements should be made at $2[m]$ above an extensive surface of green grass, shading the ground and not short of water.

Of course, this evaluation of ET_0 cannot be realistic in all the climatic situations due to some simplifications and assumptions. At this time, this is for sure one of the most accurate method to get a good estimation of that value without recurring to expensive and time consuming experiments. The FAO Penman-Monteith equation is a close, simple representation of the physical and physiological factors governing the evapotranspiration process. The data we'll need are:

- **Location**: above sea level $[m]$ and latitude (degrees north or south). These values are needed to adjust some weather parameters for the local average value of atmospheric pressure and to compute extraterrestrial radiation as well as daylight hours.
- **Temperature**: the average daily maximum and minimum air temperature in $[°C]$. We can also use the average mean daily temperature, where the other two values are not available, but this will result in some underestimation of ET_0

- **Humidity**: the average daily actual vapor pressure (e_a) in $[kPa]$ is required. In some cases it can be derived, where it's not available, from values as maximum and minimum relative humidity.
- **Radiation**: the average daily net radiation expressed in $\left[\frac{MJ}{m^2 day}\right]$.
- **Wind speed**: the average daily wind speed in $\left[\frac{m}{s}\right]$ measured at $2\ [m]$ above the ground level.

In case all or some of these parameters are missing, we can still evaluate them using other parameters. Of course, the more measure parameters we have, the more accurate our result will be.

Evaluate missing data

Sometimes we don't have all the data we need for the evaluation of ET_0. For each of the required data, we can have a different solution, trying to get that value from related parameters.

For what concern the **atmospheric pressure (P)**, we can use the following formula

$$P = 101.3 \left(\frac{293 - 0.0065z}{293}\right)^{5.26}$$

where:

- P is the atmospheric pressure $[kPa]$
- z is the elevation above sea level $[m]$

The **latent heat of vaporization (λ)** expresses the energy required to change a unit mass of water from liquid to water vapor in a constant pressure and remperature process. Over normal temperature, we can assume it constant and with the value of $2.45\ \left[\frac{MJ}{kg}\right]$

The **psychrometric constant (γ)** is given by

$$\gamma = \frac{c_p P}{\varepsilon\lambda} = 0.665 \cdot 10^{-3} \cdot P$$

where:

- γ psychrometric constant $\left[\frac{kPa}{°C}\right]$
- P is atmospheric pressure $[kPa]$
- λ latent heat of vaporization $2.45\ \left[\frac{MJ}{kg}\right]$
- c_p specific heat at constant pressure $1.013\ 10^{-3}\ \left[\frac{MJ}{kg°C}\right]$
- εratio molecular weight of water vapor/dry air 0.622

If we look at **relative humidity (RH)** we can easily evaluate with

$$RH = 100\frac{e_a}{e^o(T)}$$

where:

- RH is the relative humidity that expresses the degree of saturation of the air
- e_a actual vapor pressure. It derives from RH because it's not directly measurable
- $e^o(T)$ saturation vapor pressure

Practically, RH is measured directly with hygrometers.

To evaluate the **mean saturation pressure (e_s)** we can adopt the following equation

$$e_s = \frac{e^o(T_{max}) + e^o(T_{min})}{2}$$

where we can get $e^o(T)$ from

$$e^o(T) = 0.6108\ exp\left[\frac{17.27\ T}{T + 237.3}\right]$$

where:

- $e^o(T)$ is the saturation vapor pressure at the air temperature T $[kPa]$
- T is the air temperature $[°C]$

The **slope of saturation vapor pressure curve (Δ)** is needed for the evaluation of evapotranspiration and can be obtained from

$$\Delta = \frac{4098\left[0.6108\ exp\left(\frac{17.27\ T}{T + 237.3}\right)\right]}{(T + 237.3)^2}$$

where:

- Δ is the slope of saturation vapor pressure curve at air temperature T $\left[\frac{kPa}{°C}\right]$
- T is the air temperature $[°C]$

To evaluate the **actual vapor pressure (e_a)**, we have different methods starting from different values. We'll report the one we've used in our mathematical model, that is

$$e_a = \frac{RH_{mean}}{100}\left[\frac{e^o(T_{max}) + e^o(T_{min})}{2}\right] = \frac{RH_{mean}}{100}e_s$$

where:

- RH_{mean} is the mean relative humidity, defined as the average between RH_{max} and RH_{min}

Finally, we still need to find all the values related to radiation. For the **extraterrestrial radiation for daily periods (R_a)** we can easily use:

$$R_a = \frac{24\,(60)}{\pi}G_{sc}d_r[\omega_s \sin(\varphi)\sin(\delta) + \cos(\varphi)\cos(\delta)\sin(\omega_s)]$$

And it's expressed in $\left[\frac{MJ}{m^2 day}\right]$ and to obtain it in $\left[\frac{mm}{day}\right]$ we have to multiply it by 0.408. The meaning of the symbols in the above equation are:

- R_a is the extraterrestrial radiation $\left[\frac{MJ}{m^2 day}\right]$
- G_{sc} is the solar constant, equal to $0.082\left[\frac{MJ}{m^2 min}\right]$
- d_r is the inverse relative distance Earth-Sun
- ω_s is the sunset hour angle $[rad]$
- φ is the latitude $[rad]$
- δ is the solar declination $[rad]$

Some of the above parameters are unknown but they can be evaluated from the following equations

$$d_r = 1 + 0.033\cos\left(\frac{2\pi}{365}J\right)$$

$$\delta = 0.409\sin\left(\frac{2\pi}{365}J - 1.39\right)$$

In these 2 equations, J is the number of the day in the year between 1 (1st

January) and 366 (31st December).

$$\omega_s = \arccos(-\tan(\varphi)\tan(\delta))$$

We then need to know the **daylight hours (N)** that are given by

$$N = \frac{24}{\pi}\omega_s$$

The **solar radiation (R_s)**, if it's not given, can be evaluated as

$$R_s = \left(a_s + b_s\frac{n}{N}\right)R_a$$

where:

- R_s is the solar or shortwave radiation $\left[\frac{MJ}{m^2 day}\right]$
- n is the actual duration of sunshine $[hour]$
- N is the maximum possible duration of sunshine or daylight hours $[hour]$
- $\frac{n}{N}$ relative sunshine duration $[-]$
- R_a extraterrestrial radiation $\left[\frac{MJ}{m^2 day}\right]$
- a_s regression constant, expressing the fraction of extraterrestrial rariation reaching the earth on overcast days ($n = 0$)
- $a_s + b_s$ fraction of extraterrestrial radiation reaching the earth on clear days ($n = N$)

The **clear-sky solar radiation (R_{so})** is when ($n = N$) and it's required for computing net longwave radiation. We'll use

$$R_{so} = (a_s + b_s)R_a$$

where:

- R_{so} is the clear-sky solar radiation $\left[\frac{MJ}{m^2 day}\right]$
- $a_s + b_s$ is the fraction of extraterrestrial radiation reaching the earth on clear-sky days ($n = N$)

Then, the **net solar or net shortwave radiation (R_{ns})** is the result of the balance between incoming and reflected solar radiation given by

$$R_{ns} = (1 - \alpha)R_s$$

where:

- R_{ns} is the net solar or shortwave radiation $\left[\frac{MJ}{m^2 day}\right]$

- α albedo or canopy reflection coefficient, which is 0.23 for the hypothetical grass reference crop $[-]$
- R_s the incoming solar radiation $\left[\frac{MJ}{m^2 day}\right]$

The **net longwave radiation (R_{nl})** is proportional to the absolute temperature of the surface raised fo the fourth power.

$$R_{nl} = \sigma \left[\frac{T_{max,K}^4 + T_{min,K}^4}{2}\right] \left(0.34 - 0.14\sqrt{e_a}\right) \left(1.35 \frac{R_s}{R_{so}} - 0.35\right)$$

where:

- R_{nl} is the net outgoing longwave radiation $\left[\frac{MJ}{m^2 day}\right]$
- σ is the Stefan-Boltzmann constant $\left[4.903\ 10^{-9} \frac{MJ}{K^4 m^2 day}\right]$
- $T_{max,K}^4$ is the maximum absolute temperature during the 24-hour period $[K = °C + 273.16]$
- $T_{min,K}^4$ is the minimum absolute temperature during the 24-hour period $[K = °C + 273.16]$
- e_a actual vapor pressure $[kPa]$
- $\frac{R_s}{R_{so}}$ relative shortwave radiation (limited to ≤ 1.0)
- R_s is the measured or calculated solar radiation $\left[\frac{MJ}{m^2 day}\right]$
- R_{so} is the calculated clear-sky radiation $\left[\frac{MJ}{m^2 day}\right]$

Finally, the **net radiation (R_n)**, that is the radiation cleared from all the dispersions, can be evaluated as

$$R_n = R_{ns} - R_{nl}$$

For what concern the **soil heat flux (G)** , there are complex models to describe it, we have chosen a simple calculation procedure for long time steps, based on the idea that soil temperature follows air temperature

$$G = c_s \frac{T_i + T_{i-1}}{\Delta t} \Delta z$$

where:

- G is the soil heat flux $\left[\frac{MJ}{m^2 day}\right]$
- c_s is the soil heat capacity $\left[\frac{MJ}{m^3 °C}\right]$
- T_i is the air temperature at time i $[°C]$
- T_{i-1} is the air temperature at time $i-1$ $[°C]$

- Δt is the length of time interval $[day]$
- Δz is the effective soil depth $[m]$

For what concern the wind speed, we'll use data as presented in the next chapter, but we could also evaluate it with the following equation

$$u_2 = u_z \frac{4.87}{\ln(67.8z - 5.42)}$$

where:

- u_2 is the wind speed at $2\ [m]$ above ground surface $\left[\frac{m}{s}\right]$
- u_z is measured wind speed at z meters above ground surface $\left[\frac{m}{s}\right]$
- z height of measurement above ground surface of u_z $[m]$

Now we're able to evaluate all the data we need, and we can add it to the Penman-Monteith equation above written and there under reported:

$$ET_0 = \frac{0.408\Delta(R_n - G) + \gamma \frac{900}{T + 273} u_2 (e_s - e_a)}{\Delta + \gamma(1 + 0.34u_2)}$$

3. Database and Technology

Before proceeding to explain how the code have been written, based on the theoretical equations we've just presented, we need to introduce the technology used to develop the WaterGrab.Zone project.

The user interface of the website is merely a website, that runs on a Web Server suing LAMP technology (Linux, Apache MySql and PHP).

The data are saved into a relational Database that is MySql, as it will be presented in the following paragraphs.

The code has been mainly written in 5 different programming languages:

- HTML: for the graphical part of the user interface
- JavaScript: for the graphical effects of the user interface
- PHP: to do all the calculations and manage all the dynamic part of the page
- Octave: to manage all the theoretical equations, more in detail to evaluate ET_0 and ET_C. This programming language is very similar to Matlab, with the difference that the interpreter can be easily run on a shell in a Linux server ambient, and it's code is open source (free)
- MySQL: to communicate with the database

The data we need have been taken from different sources. Some of these data require to be update daily, and this is reached with ad-hoc made daily routines.

Climate data

First of all we need to find out, with monthly frequency:

- T_{max} $[°C]$ max temperature
- T_{min} $[°C]$ minimum temperature
- RH_{mean} $[\%]$ relative humidity
- u_2 $\left[\frac{m}{s}\right]$ wind speed

- $N \left[\frac{hours}{day}\right]$ daylight sun hours
- $p \left[\frac{mm}{month}\right]$ precipitation

Of course we cannot have these data with monthly frequency in any part of the world. We can have them in a few points, better known as climate stations. From the FAO website, we've downloaded the *Climwat 2.0* tool, designed for *CropWat* software. In this tool, there's a big official database with 4251 stations in the world.

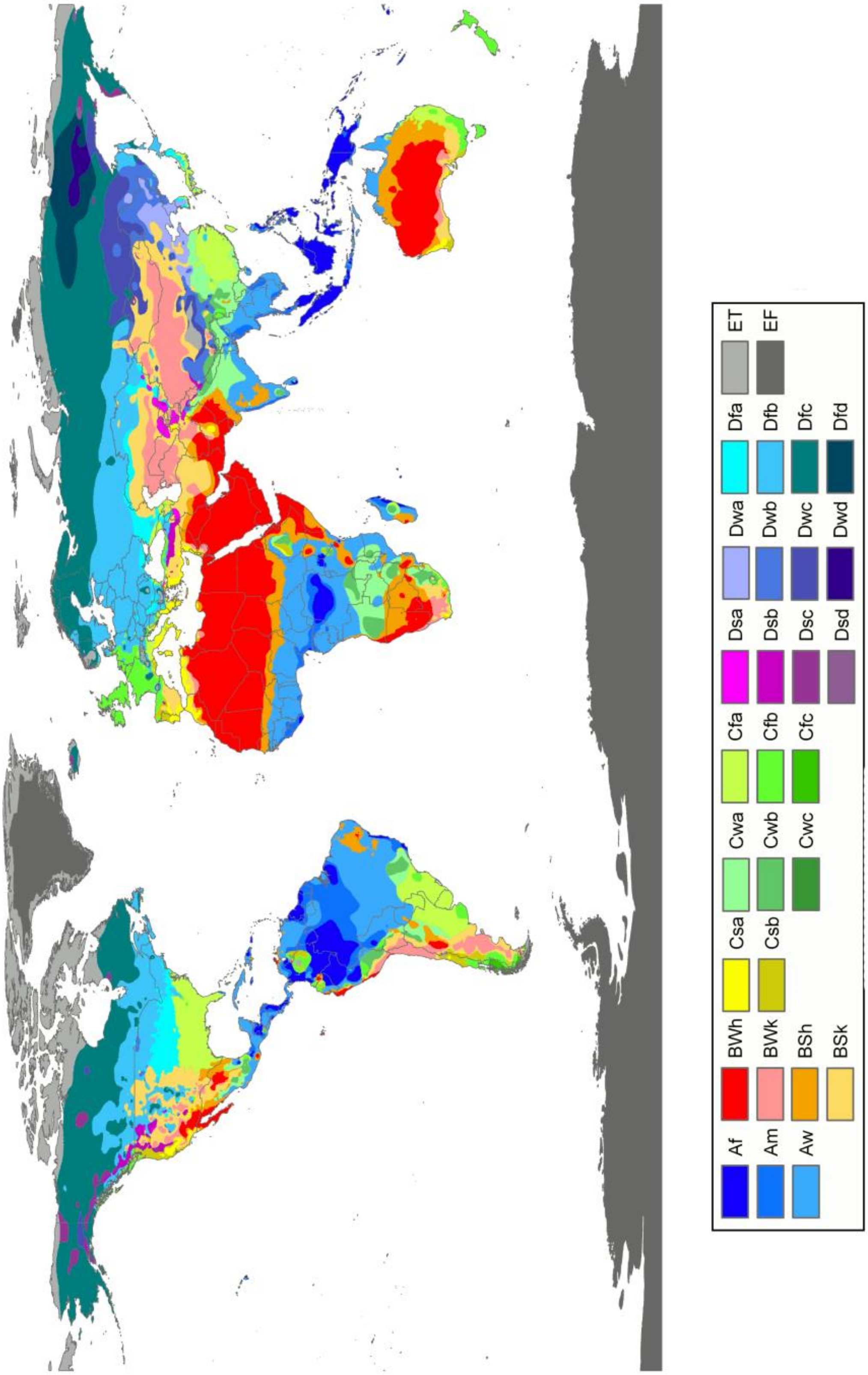
For each station we have all the data we need, divided by month. For each station, of course, we have also a name, a location (in terms of latitude and longitude) and an altitude. An example of station data it's given in the following table, showing the station of *Milano*.

ID: *2506*	**Name:** *MILANO*	**Latitude:** $45.46°$
		Longitude: $9.18°$
		Altitude: $121\ [m]$

$Month$	$T_{max}\ [°C]$	$T_{min}\ [°C]$	$RH_{mean}\ [\%]$	$u_2 \left[\frac{m}{s}\right]$	$N \left[\frac{hours}{day}\right]$	$p \left[\frac{mm}{month}\right]$
1	4.5	-0.2	88	0.7	0.79	44
2	7.6	1.9	81	0.7	2.13	60
3	13.1	5.9	77	1.1	3.33	77
4	18.2	9.7	73	1.1	4.75	94
5	23.1	13.9	73	1.1	5.72	76
6	26.5	17.2	72	1.1	6.38	118
7	29.1	19.6	70	0.7	7.78	64
8	27.6	18.9	74	0.7	6.76	91
9	23.8	16.1	77	0.7	5.04	69
10	17.2	11	84	0.4	3.25	125
11	10.4	6.1	88	0.7	0.97	122
12	5.9	1.7	91	0.4	0.37	77

Then all the world has been divided in climatic regions, according to Köppen division.

The **Köppen Climate Classification System** is the most widespread system used to classify the climates of places on our planet. The system was developed German climatologist *Wladimir Köppen* who divided the world's climates into several major categories based upon general temperature profile related to latitude. It is an empirical system based on observable features and it classifies a location’s climate mainly using annual and monthly averages of temperature and precipitation. Each region is marked with a series of letters, as we can see in the following image.


Af
Am
Aw
BWh
BWk
BSh
BSk
Csa
Csb
Cwa
Cwb
Cwc
Cfa
Cfb
Cfc
Dsa
Dsb
Dsc
Dsd
Dwa
Dwb
Dwc
Dwd
Dfa
Dfb
Dfc
Dfd
ET
EF

Basically there are major categories, that are those with capital letters:

- **A**: tropical moist climate
- **B**: dry climates
- **C**: moist mid-latitude climates with mild winters
- **D**: moist mid-latitude climates with cold winters
- **E**: polar climates

Then the second letters have different meaning, that are:

- Lowercase letters **f**, **w** and **s** are used to distinguish precipitation patterns and are only applicable to **A**, **C** and **D** climates
- Uppercase **W** and **S** identify **desert** (arid) or **steppe** (semiarid) climate subtypes for the **Dry Climates**(**B**) major category
- For the **Polar Climates** (**E**), **F** and **T** distinguish whether the site is covered by permanent ice fields and glaciers or free of snow and ice during the summer season

B, C and **D** have even a third letter, that is:

- In **B** climates, the letter **h** identifies a subtropical location where average annual temperature is above $18\ [°C]$. Cooler mid-latitude **Dry Climates** are distinguished with a lowercase **k**
- For **C** and **D** climates the letters **a, b, c** and **d** are used to distinguish different monthly temperature characteristics

Soil data

As we said before, also the kind of soil plays an important role in the crop growing. A soil with a big percentage of clay will keep more water than a soil with prevalence of sand.

The data for the soil are taken from the HWSD database (Harmonized World Soil Database) that’s open source and everyone can download it at the following page

http://webarchive.iiasa.ac.at/Research/LUC/External-World-soil-database/HTML/

We’ve divided the whole world into points, with a step of $0.00833°$ of latitude (the same for longitude) between points. Since we’ve considered only the dry land, for the 90% of these lands the distance between the points it's around $900\ [m]$.

For each terrain we've taken a lot of different values. We have the average soil composition of each point for the first $30\ [cm]$ and also for the first $100\ [cm]$. The composition is merely given in $[\%]$ of sand, silt and clay.

These data are the most memory consuming inside our project, since we have around 220 million points to gain a really high precision.

Crop data

A big part of our database is occupied by crop parameters. For each crop we have to define the following parameters:

- $K_{C,ini}$ the value of K_C needed in the first stage of the crop to evaluate ET_C from ET_0
- $K_{C,mid}$ the value of K_C needed in the second stage of the crop to evaluate ET_C from ET_0
- $K_{C,end}$ the value of K_C needed in the final stage of the crop to evaluate ET_C from ET_0
- $InitialStage\ [days]$ how long is the initial stage of the crop
- $DevStage\ [days]$ how long is the development stage of the crop
- $MidStage\ [days]$ how long is the middle stage of the crop
- $LateStage\ [days]$ how long is the late stage of the crop
- $DatePlant\ [dats]$ the day of the year in which we should plant the crop (it's the number of days since the beginning of the year)
- $RootInit\ [m]$ the depth of the root in the initial stage of the crop
- $RootFinal\ [m]$ the depth of the root in the final stage of the crop
- $pStandard\ [-]$ coefficient to get RAW from TAW, where RAW is the *Readly Available soil Water* and TAW is the *Total Available soil Water*.

These values can be found inside the *Water Footprints of Nations*, a document published in November 2004 by UNESCO-IHE (Institute of Water Education) with authors *A.K. Chapagain* and *A.Y. Hoekstra*. The document can be downloaded from the following link

http://doc.utwente.nl/77203/2/Report16Vol2Appendices.pdf

The document is in our interest for the table in *Appendix VI* where we'll find the different crops with all the values above mentioned. As we can see, there are no present the Köppen climate regions, but 7 regions that are:

- Tropics
- Subtropiics summer rainfall
- Subtropics winter rainfall
- Oceanic temperate
- Sub-continental temperate and continental temperate
- Sub-continental boreal, continental boreal and polar/artic
- Deserts

After a brief search, we came out with the following conversion table. For each of the 7 climate regions, we proved the corresponding Köppen value:

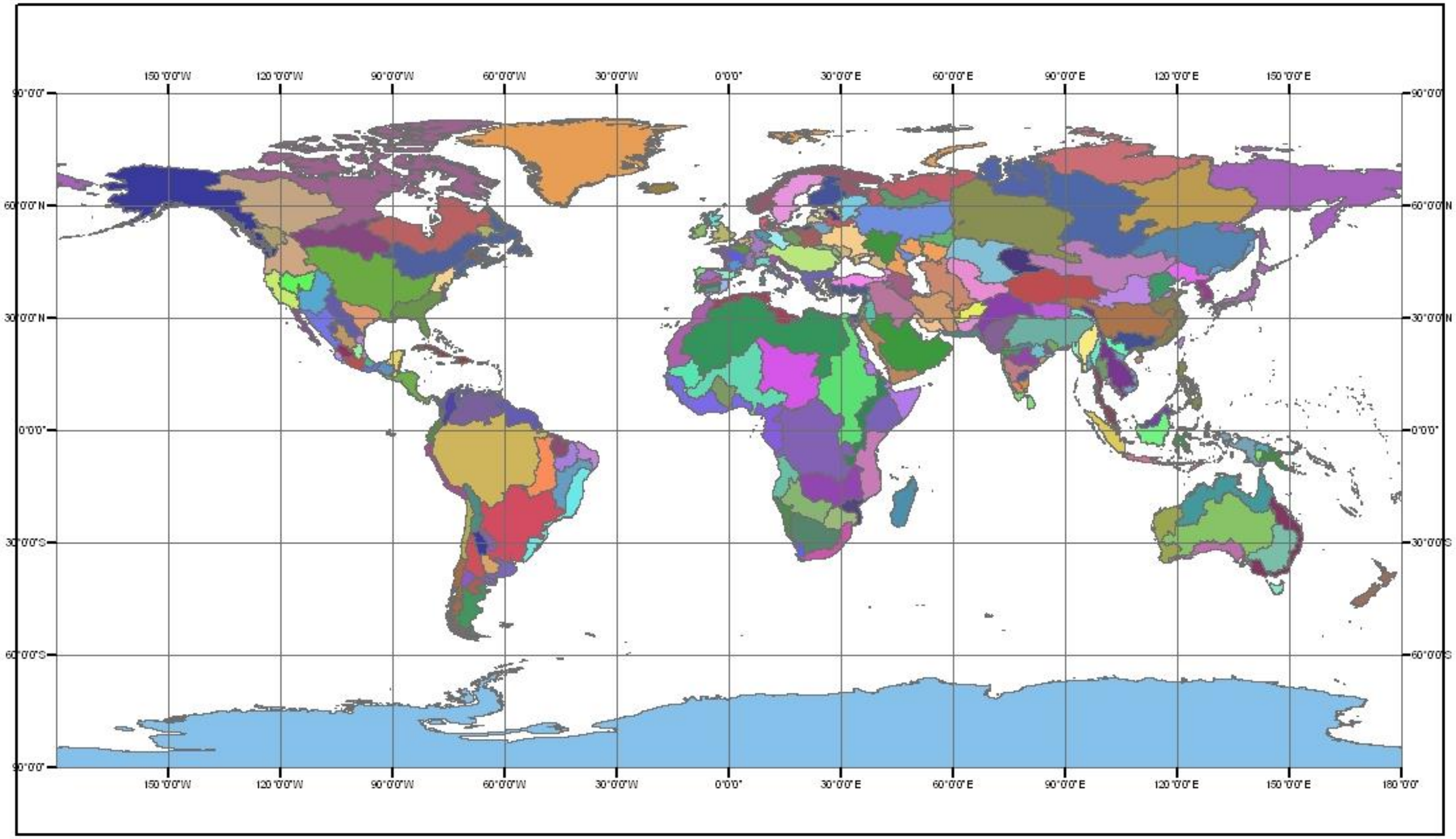
Climate Region	**Corresponding Köppen Climate Region**							
Tropics	Af	Am	Aw					
Subtropics summer rainfall	Cwa	Cfa						
Subtropics winter rainfall	Csa	Csb	Csc					
Oceanic temperate	Cwb	Cwc	Cfb	Cfc				
Sub-continental temperate and continental temperate	Dsa	Dsb	Dsc	Dsd	Dwa	Dwb	Dwc	Dwd
Sub-continental boreal, continental boreal and polar/arctic	Dfa	Dfb	Dfc	Dfd	ET	EF		
Deserts	BWh	BWk	BSh	BSk				

Spatial data

Finally, we need to define the hydrological basins, that are areas where all the surface water from rain, melting snow, or ice converges merges to the same point. We've defined macro hydrobasins in the world, that can be easily downloaded from the *FAO AquaMaps* online tool, accessible at the following URL

http://www.fao.org/nr/water/aquamaps/

Major hydrological basins of the world



We need to define them to make sure we'll use the correct climate stations, not to use data of a station that will never influence a point since the precipitation in that area will refer to another hydrological basin.

We've imported them through GIS into our database as geometrical polygons, saving all the borders as well as the centroid of that polygon.

Deals

Thanks to the *LandMatrix* project (http://www.landmatrix.org/en/), we're able to get all the deals regarding terrains in the world. These deals are the contract done by governments or private companies that want to buy large fields for cultivate purposes. These deals will show us a lot of information, such as:

- The location of the deal in the world;
- Who is going to buy that land;
- Which are the intentions of the buyer;
- The status of the deal (completed, still in discussion, etc)
- The size of the deal;
- The nature of the deal (lease, outright purchase, etc)
- The crop they're going to plant there

For what concern the intentions, we'll have basically one or more of the following intentions for each deal:

Intentions
Agriculture (general)
Biofuels
Conservation
Food crops
For carbon sequestration
For wood and fiber
Forest
Industry

Livestock
Non-food agricultural commodities
Renewable Energy
Tourism

These deals are update daily into our system thorough an automated process.

4. Our code for ET_0 and ET_C

Now that we have all the data to deal with Penman-Monteith equations, we can start presenting the code that we've used to reach these steps.

First of all, the core of our code is written, as said in Chapter 3, in Octave, an open source version of MatLab. Octave itself won't be able to do a lot, and the two main script of Octave are governed by PHP scripts.

We'll start analyzing the first script, that will evaluate ET_0, to then see the second script, that will allow us to evaluate ET_C. Finally, we will still need to make a complete water balance and see if our values are reasonable of we've made some mistakes in our calculations.

Before starting with ET_0, just a brief introduction in the structure of WaterGrab.Zone project. All the project is object-oriented programmed, that is for each task there's a specific object, or class, that will be able to solve that.

Evaluate ET_0

The class that will manage the operation of evaluating ET_0 is called *et0Calculator* . This class needs, in our point of calculation, all the data seen in the Penman-Monteith equation:

- Temperature max and min;
- Relative humidity;
- Wind speed;
- Hours of sun;
- Precipitation;
- Latitude, longitude and altitude.

Plus other parameters functional to the evaluation, such as:

- connection to database, to read all the data we need, such as the climate stations;

- *spaceManager* object, that’s the object that manage all the spatial calculations;
- the stations we want to use for the calculations. In case we won’t make any preferences, the object will use all the available stations inside the hydrological basin

We start setting up the object with:

- latitude and longitude of the point;
- the climate stations we want to use (if we have preferences);

So now the object is ready and can evaluate ET_0. First of all we’ll read all the values of climate stations chosen by the user or inside our hydrological basin. These values will give us the data needed by Penman-Monteith equation. We then need to find those values in our point, and we can easily do that with an inverse weighted average: the closet the station is, the more influence will have.

These values will be printed into a temporary CSV file, that’s the file used by our Octave script. Once the file is ready, our object will call *ET0_month.m*, the script that will run the simulation. Here is the code:

```
%  ================================================
%            ET0 FOR MONTHLY MEAN DATAS
%  ================================================

%args passed are:
%- argv(){1}  -->   name of the file with data

% The file we’ll read has:
% - Tmin  [Array]
% - Tmax  [Array]
% - Wind speed  [Array]
% - Sun hours  [Array]
% - J, day in which we want to make the calculation (15 if it’s middle of the month)  [Array]
% - altitude
% - latitude
filename = strcat('datas/', argv(){1}, '.csv');
values = csvread(filename);
Tmin = values(:, 1);   %  [°C]
Tmax = values(:, 2);   %  [°C]
u2 = values(:, 3);     %  monthly average daily wind speed  [m/s]
n = values(:, 4);      %  monthly average sunshine duration [hours/day]
RH = values(:,5);       % RH [%]
J = values(:, 6);        % number of day since beginning of year
```

```
z = values(:, 7);           % altitude [m]
latitude = values(:, 8);  % latitudine [°N]



% === PARAMETERS ===
Tmean = (Tmax + Tmin)./2;   %[°C]  -  eq. 9
delta = (4098.*(0.6108.*exp((17.27.*Tmean)./(Tmean+237.3))))./(Tmean+237.3).^2;
%  [KPa / °C]  - eq. 13
P = 101.3 .* ((293 - 0.0065.*z)./293).^5.26;   %[KPa]  -  eq. 7
gamma = 0.665.*10.^-3.*P;   %[KPa / °C]  -  eq. 8

%I'm making the Tmean array. We have always 12 months of datas (csv with 12
lines!)
Tmonth_im = zeros (length(Tmean), 1);
Tmonth_ip = zeros (length(Tmean), 1);
for i=1:length(Tmean)
    if i==1
        Tmonth_im(i) = Tmean(length(Tmean));
        Tmonth_ip(i) = Tmean(i+1);
    elseif i==12
       Tmonth_im(i) = Tmean(i-1);
       Tmonth_ip(i) = Tmean(1);
    else
        Tmonth_im(i) = Tmean(i-1);
        Tmonth_ip(i) = Tmean(i+1);
    end
    Tmonth_ip(i)=-273;
end

% === VAPOUR PRESSURE DEFICIT ===
e0_tmax = 0.6108.*exp((17.27.*Tmax)./(Tmax+237.3));   %[KPa]  -  eq. 11
e0_tmin = 0.6108.*exp((17.27.*Tmin)./(Tmin+237.3));   %[KPa]  -  eq. 11
es = (e0_tmax+e0_tmin)./2;   %[KPa]  -  eq. 12
ea = es.*RH/100;   %    [KPa]   -  eq. 18

% === RADIATION ===
dr = 1+0.033.*cos(2.*pi.*J./365);  %  Inverse relative distance Earth-Sun  -
eq. 23
fi = latitude .* pi./180;   %  grads into radiants  [rad]  -  eq. 22
delta_min = 0.409.*sin((2.*pi.*J./365)-1.39);  % [day]  -  eq. 24
omega_s = acos(-tan(fi).*tan(delta_min));   % [rad * day]  -  eq. 25
Ra =
60.*24./pi.*0.0820.*dr.*(omega_s.*sin(fi).*sin(delta_min)+cos(fi).*cos(delta_min
).*sin(omega_s));   % [MJ m-2 day-1]  -  eq. 21

N = 24.*omega_s./pi;  %  daylight hours -  eq. 34
nN = n./N;
Rs = (0.25+0.5.*nN).*Ra;   % Solar Radiation [MJ m-2 day-1]    -  eq. 35
                           % 0.25+0.5 it's the fraction of extraterrestrail
radiation that reaches earth in days with clear-sky (ove n=N)
Rs0 = (0.25+0.5+2.*10.^-5.*z).*Ra;  %Clear-Sky sloar radiation (Rs0)  [MJ m-2
day-1]  -  eq. 36
RsRs0 = Rs./Rs0;
Rns = 0.77.*Rs;   %net solar radiation   [MJ m-2 day-1]  -  eq. 38

TmaxK = Tmax + 273.16;  % [K]  -  eq. 39(note)
TminK = Tmin + 273.16;  % [K]  -  eq. 39(note)
Rnl = 4.903.*10.^-9.*((TmaxK.^4 + TminK.^4)./2).*(0.34-
0.14.*sqrt(ea)).*(1.35.*RsRs0-0.35);   %  net longware radiation   [MJ m-2 day-
1]  -  eq. 39
```

```
Rn = Rns - Rnl;  %  Net radiation    [MJ m-2 day-1]   -   eq. 40

G = zeros(length(Tmean), 1);
for i=1:length(G)
    if Tmonth_ip(i) < -200
        G(i) = 0.14.*(Tmean(i) - Tmonth_im(i));  %Soil heat flux   [MJ m-2 day-
1]  -   eq. 44
    else
        G(i) = 0.07*(Tmonth_ip(i) - Tmonth_im(i));  %Soil heat flux   [MJ m-2
day-1]  -   eq. 43
    end
end


% === GRASS REFERENCE EVAPOTRANSPIRATION ===
ET0 = (0.408.*delta.*(Rn - G) + gamma.*900./(Tmean+273).*u2.*(es-ea)) ./
(delta+gamma.*(1+0.34.*u2));   %[mm/day]  -  6
filename = strcat('datas/', argv(){1}, '_res.csv');
fid = fopen(filename,'w');  %wt write in text mode
fprintf(fid,'%f\n',ET0);
fclose(fid);
```

Reading the input file, we apply all the equations we've seen in Chapter 3, to finally get the ET_0 values (one value for each month). These 12-values array is then saved into a file, so that the object will be able to read the results of this calculation.

Back to the object, we then have to open the file create by Octave script, read these values and store in our memory variables, ready to be used by ET_C calculation.

Evaluate ET_C

We're now ready to use, with the same logic as for ET_0, the object *etCCalculator*. This object will require an *et0Calculator* object with all its results inside. Remembering how we evaluated ET_0, here we can simply evaluate ET_C with the following equation:

$$ET_C = ET_0 \cdot K_c$$

Remembering that:

- We have a monthly distribution of precipitation, instead we need a daily distribution;
- K_c depends from crop and from stage period (so we will need daily values for this parameter)

We then need to initialize this object with also the following parameters:

- the crop we want to simulate (you can find a full list in Appendix 1);
- the date of planting inside the year.

Once we have the ET_0 calculation, we then need to split these monthly values into daily values. I'll do the same with the different values of K_c. For what concern precipitation, we have monthly values and we can recreate different raining scenarios. For a mere purpose of simulation, we followed the *FAO Cropwat* solution that concentrates rain into 6 days in a month, that are the *3 and *7 days (3, 7, 13, 17, 23 and 27). So we'll divide the value of monthly precipitation into 6 equal values.

Similar to what we've done for ET_0, we'll prepare two files that will be read by the Octave script, with all the data required: one with the 365 values of ET_0, another one with the crop specific values. The Octave script (*ETc_month.m*) will read these files and calculate the ET_C day by day.

The code is the following;

```
%  =================================================
%       ETc DERIVED FROM ET0 WITH MONTHLY DATAS
%  =================================================

%  REFERENCEs are to FAO56 PDF

%args passed are:
%- argv(){1}  -->   name of the file with the data needed

% We'll read two files.
% The first one is a list of 365 values for ET0
% The secondi is a list of values with the following order
% + KCini (initial KC)
% + KCmid (middle KC)
% + KCend (final KC)
% + InitialStage (duration of initial Stage)
% + DevStage (duration of development Stage)
% + MidStage (duration of middle Stage)
% + LateStage (duration of Late Stage)
nomeCodice = argv(){1};

filename = strcat('datas/', nomeCodice, '.csv');
values = csvread(filename);
ET0 = values(:, 1);   %  [mm/day]

filename = strcat('datas/', nomeCodice, '_cropParam.csv');
values = csvread(filename);
KCini = values(1, 1);  %  [-]
KCmid =  values(1, 2);  %  [-]
KCend =  values(1, 3);  %  [-]
InitialStage =  values(1, 4);  %  [days]
DevStage =  values(1, 5);  %  [days]
```

```
MidStage =  values(1, 6);  %  [days]
LateStage =  values(1, 7);  %  [days]
StartDate = values(1, 8);    %  [days]   day in which we start evaluate the crop
plot
RootInit = values(1, 9);    %[m]
RootFinal = values(1, 10);  %[m]


% === CALCULATE ETc ===
       %[mm/day]

Kc = zeros(rows(ET0),1);
root = zeros(rows(ET0),1);
stage = repmat('n',rows(ET0),1);
ETc = zeros(rows(ET0),1);
Offset = StartDate-1;

% if it's in INITIAL STAGE
DevStageTOT = InitialStage + DevStage;
for i = 1:InitialStage
    if (Offset + i) >= 366
        Offset = -(i-1);
    end
    ETc(Offset + i) = ET0(Offset + i)*KCini;
    Kc(Offset + i) = KCini;
    stage(Offset + i) = 'i';
    root(Offset + i) = (i*(RootFinal-RootInit)/(DevStageTOT))+RootInit;
end

% if it's in DEVELOPMENT STAGE
for i = InitialStage:DevStageTOT
    if (Offset + i) >= 366
        Offset = -(i-1);
    end
    KCtmp = ((KCmid - KCini) * (i-InitialStage) / DevStage) + KCini;
    ETc(Offset + i) = ET0(Offset + i)*KCtmp;
    Kc(Offset + i) = KCtmp;
    stage(Offset + i) = 'd';
    root(Offset + i) = (i*(RootFinal-RootInit)/(DevStageTOT))+RootInit;
end

% if it's in MIDDLE STAGE
MidStageTOT = DevStageTOT + MidStage;
for i = DevStageTOT:MidStageTOT
    if (Offset + i) >= 366
        Offset = -(i-1);
    end
    ETc(Offset + i) = ET0(Offset + i)*KCmid;
    Kc(Offset + i) = KCmid;
    stage(Offset + i) = 'm';
    root(Offset + i) = RootFinal;
end

% if it's in LATE STAGE
LateStageTOT = MidStageTOT + LateStage;
for i = MidStageTOT:LateStageTOT
    if (Offset + i) >= 366
        Offset = -(i-1);
    end
    KCtmp = ((KCmid - KCend) * (LateStageTOT-i) / LateStage) + KCend;
```

```
    ETc(Offset + i) = ET0(Offset + i)*KCtmp;
    Kc(Offset + i) = KCtmp;
    stage(Offset + i) = 'l';
    root(Offset + i) = RootFinal;
end


%write the result into a file
filename = strcat('datas/', nomeCodice, '_res.csv');
fid = fopen(filename,'w');  %wt to write in text mode
for i = 1:rows(ET0)
    fprintf(fid,'%f,a,%f,a,%c,a,%f\n',ETc(i,1),Kc(i,1),stage(i,1),root(i,1));
end
fclose(fid);
```

Water Balance Equation

At this stage of calculation, we have the value of ET_C in each day of our crop. We still don't know how much water we have to provide by irrigation, and when.

The last object we'll introduce to complete our calculation, is *balanceEqCalculator* class. This class will require a complete *etCCalculator* object and it won't run any Octave script, but it'll all be done inside PHP ambient.

Basically in this evaluation, we'll run a *soil water balance* utilizing the following parameters:

- I irrigation made by human
- P precipitation
- RO surface run off
- DP deep percolation of exceeding water
- ET evapotranspiration of the system
- SB soil balance (water present inside the soil)

The equation is the following:

$$SB = I + P - RO - DP - ET$$

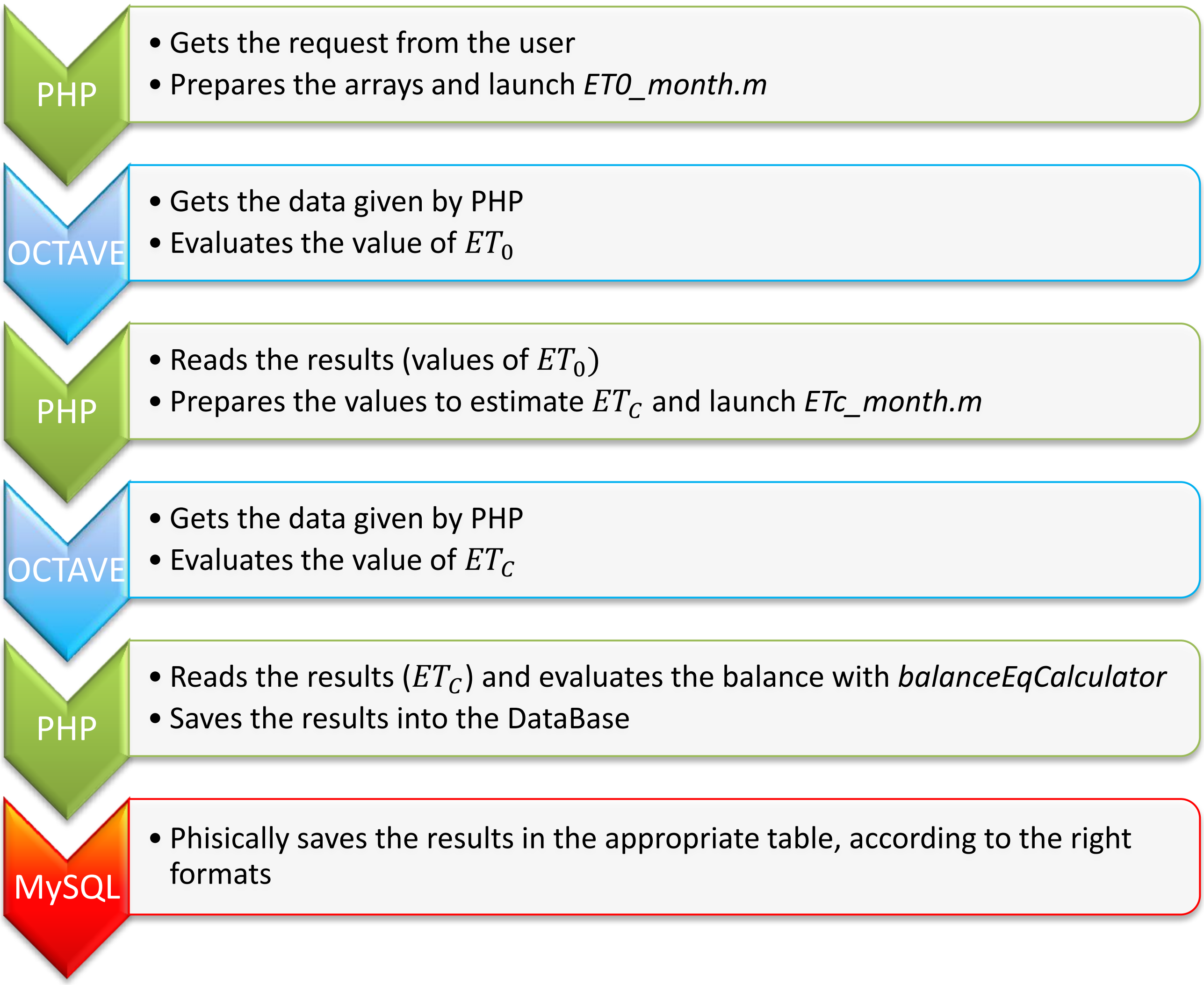
Of course, here, our goal is to value the irrigation needed to avoid water stress for the crops. We have water stress when $SB > RAW$, where RAW is the *Readily available water*. The *Readily available water* is calculated thanks to the soil: this is why we needed to add the soil data inside our database. RAW basically indicate an height into the soil where, if we have water, we can use that water for our crop.

Till this point we have evaluated ET_C, so the water needed by the crop. Now the soil type plays an important role, storing the water needed by the plant and given even when there's no rain.

To use the above equation (balance equation) we'll use the following algorithm:

- if we have more water than the maximum storable into the ground, we'll have deep percolation ($DP > 0$), i.e. we'll remove the excess of water. In this case we won't have irrigation, and it's typically during a period with precipitation.
- if we have less water than the maximum storable into a system, but the RAW value is not reached, we have enough water inside our system to feed the plan without incurring into water stress. No irrigation is needed, neither deep percolation is expected.
- if we have less water than the RAW value, we need to give water artificially to the plant. We can easily do this with irrigation: that's the case where we use an irrigation input. We won't just give the minimum amount of water to have more than RAW value in the terrain, but we will give water till we fill all the storable space in the porous medium, without having waste (avoiding deep percolation).

All the results will be saved inside our MySQL database, following the algorithm logic that's sum up in the next image.



PHP
- Gets the request from the user
- Prepares the arrays and launch *ET0_month.m*

OCTAVE
- Gets the data given by PHP
- Evaluates the value of ET_0

PHP
- Reads the results (values of ET_0)
- Prepares the values to estimate ET_C and launch *ETc_month.m*

OCTAVE
- Gets the data given by PHP
- Evaluates the value of ET_C

PHP
- Reads the results (ET_C) and evaluates the balance with *balanceEqCalculator*
- Saves the results into the DataBase

MySQL
- Phisically saves the results in the appropriate table, according to the right formats

Comparison data

Before producing the user interface, we run several tests to make sure our procedure is correct and won't stuck.

The first step is to simulate the ET_0 values. Since we've followed the FAO suggested model, we have a lot of references of previous case of studies, also with correlation with real data.

For what concern ET_0 the values obtained have an average of variation of 0.1% compared to the same simulations run with *Cropwat* software. For the climate stations we have also reference (measured) ET_0 values in these points. For the 90% of these stations, the difference of ET_0 is less than 2%, and for the other 10% is less than

3%, that means a great correlation with real data.

Once we move to ET_C and also to the soil water balance, we have a remarkably difference for simulations involving rice. This was expected, since rice typically grow in wetland areas, where there's always a layer of water and the surface evaporation, as well as deep percolation, plays a bigger role. This implies a higher utilize of water compared to the values obtain using our model.

This last aspect is emphasize inside the user interface part of the project, by underlying that the water needed by rice is underestimate.

5. WaterGrab.Zone project

Now that we have all the code ready to evaluate the water need, we can built the user interface that will manage all the process: from choosing a location and a crop, to visualize the values, exporting or reading them again in the future.

Let's start presenting the structure, with a brief look to all its features, to see it working with a couple of examples and data output. Finally, there are some features that can be added in future developments.

Website structure

The website it's very simple, and we can divide it in the following areas:

- **Profile page**: where the user can find his profile and all the simulations he has done in the past.
- **Simulation page**: the page where a user can set up his simulation, choosing all the parameters we'll see later
- **Result page**: here we can see the results of our simulation, with all the chart needed to make the results even more clear.

The profile page allows us to:

- see our profile data;
- look into our simulations done in the ast and, for each simulation:
    - see the results;
    - download the results into an Excel file;
    - delete the simulation

The simulation page is divided into 4 steps:

1. We need to give a name to the simulation, so that we can easily find

inside our profile later on;

2. we have to set the location of our point, giving all the requested spatial information:
    a. latitude and longitude of the point can be found writing directly them inside, moving the red marker on the map or searching for a specific address using the search box on the top left of the map
    b. the surface in hectares, so that we cannot just get back a value in $[mm]$, but also a value in $[m^3_{water}]$.
    c. The number of climate stations we want to use. Once we've chosen a number (the default is 6), we can then decide if we want to cancel some of these, maybe with a criteria of heights (if a station is lower than my point, I can cancel it). We can do all that management with the orange "Manage stations" button.
3. We need to choose the crop, writing the name in the input field and select the corresponding crop in the list menu. It's possible also to translate the name of the crop from any language, choosing the "translate" green button right under.
4. We can finally choose a custom date of planting, otherwise the simulation will use the default FAO suggested date.

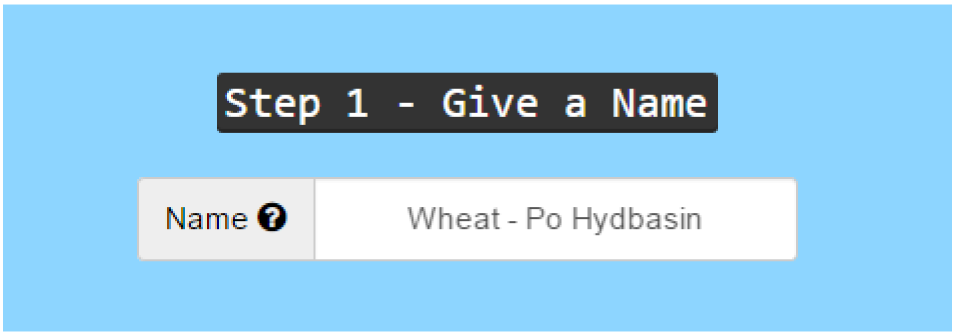
In step number *2*, in the map, we can also see (with a red line shape) the hydrological basin in which our point is. Of course, the climate stations will be chosen inside the hydrological basin, and never outside.

Finally, we can simulate an existing deal, without entering by hand all the data needed every time. We just need to click on "*choose an existing deal*" right under the title "*Set your simulation*" and, there, select an existing deal, searching inside the given table with all the recorded deals.

How does it work?

To start, we need to open the simulation page. Here we'll make an example to start. We'll plant *Wheat* close to *Parma*, so in the *Po hydrological basin*. We'll have a surface of *50 hectares* and we won't change the default planting date.

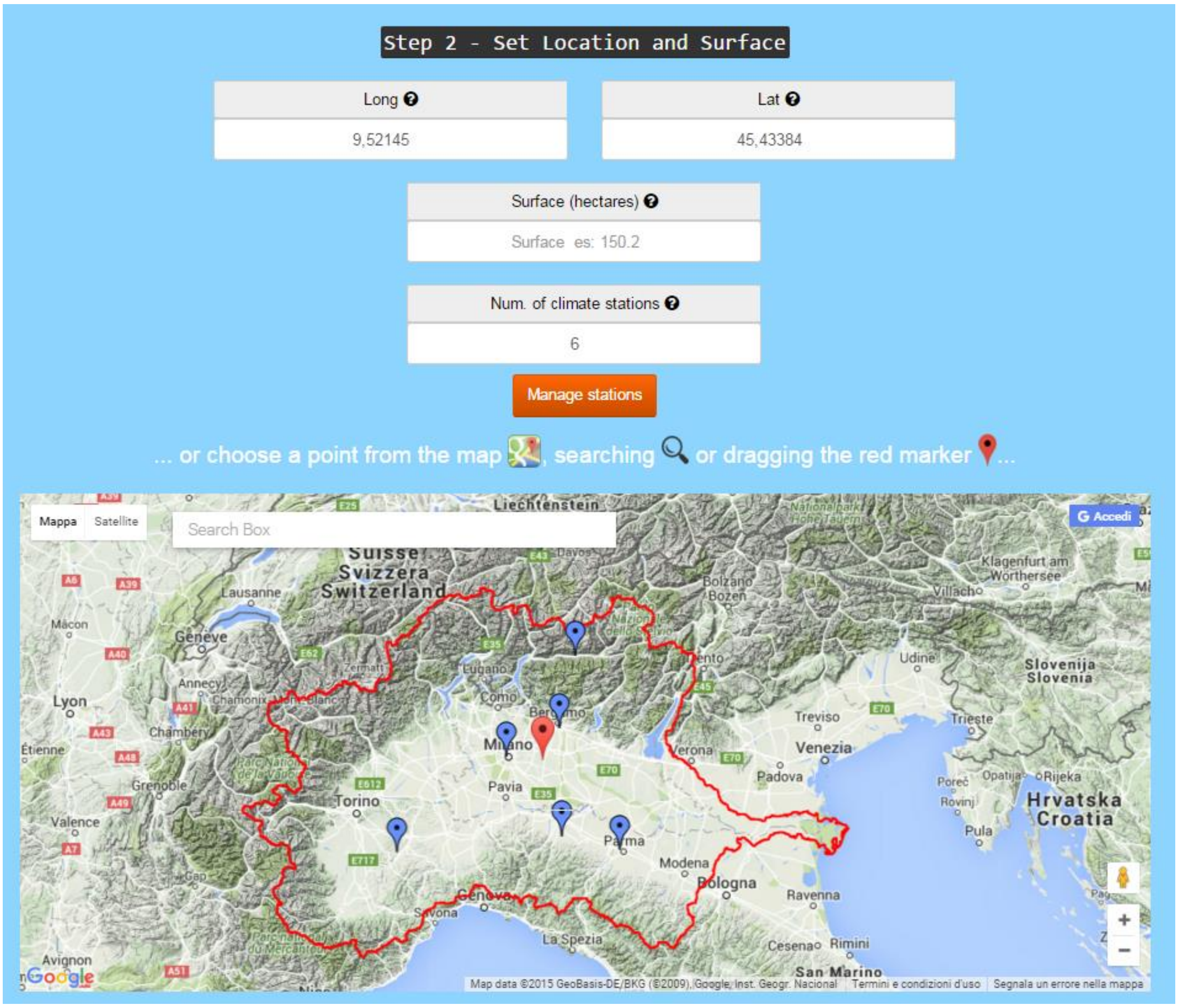
Starting by giving a name, let's call it "*Wheat – Po Hydbasin*"



Step 1 - Give a Name

Name: Wheat - Po Hydbasin

We can now move to step 2, and, as we can see, the tool place a default location very close to Milan. We have, to start, the following situation:



Step 2 - Set Location and Surface

Long: 9,52145

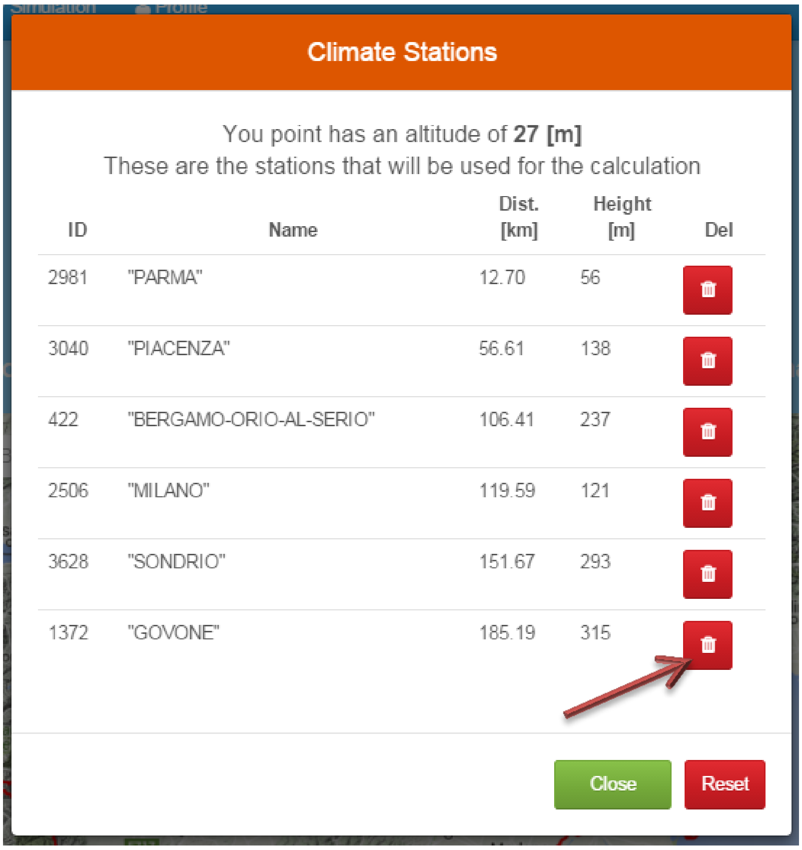
Lat: 45,43384

Surface (hectares): Surface es: 150.2

Num. of climate stations: 6

We need to put our marker close to Parma, and we can simply drag and drop it on the map. Once the map finished loading the new climate stations, we have the following map situation:

If we want to exclude from the calculation the station of *GOVONE*, that's the one on the left, we simply click on "*Manage stations*" orange button, and we remove that station from the list that wil appear.



Climate Stations

You point has an altitude of **27 [m]**
These are the stations that will be used for the calculation

ID	Name	Dist. [km]	Height [m]	Del
2981	"PARMA"	12.70	56	🗑
3040	"PIACENZA"	56.61	138	🗑
422	"BERGAMO-ORIO-AL-SERIO"	106.41	237	🗑
2506	"MILANO"	119.59	121	🗑
3628	"SONDRIO"	151.67	293	🗑
1372	"GOVONE"	185.19	315	🗑

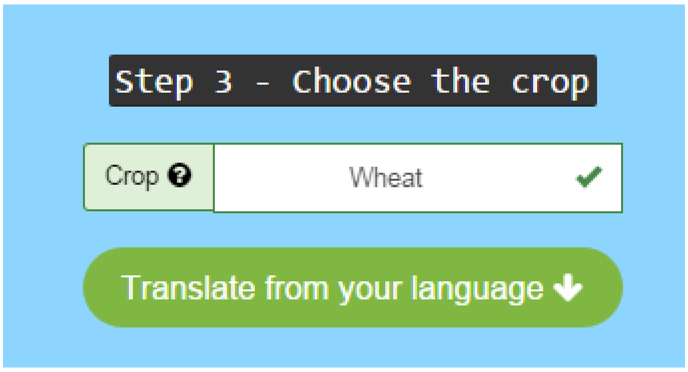
Remove it's very simple, since we just need to click on the trash button on the line we want to remove. As you can see, in that page we can see the height of our point (in our case, $27\,[m]$) and the height of the stations, that are listed from the closest to the farther.

If we cancel the wrong station, we can always press "*Reset*" to resume the original situation of the stations. Once we've finished, "*Close*" will allow us to close the window and go back to the simulation page.

Once back to the simulation page, we see that the *GOVONE* climate station is no more present, and we have only the 5 closest stations to Parma.

Don't forget to add the surface for the simulation. We said we want to use a land of *50 hectares*, so just put 50 in the corresponding field.

Step number 3, we have to choose the crop: just need to write "*Wheat*" and select the option that will appear.

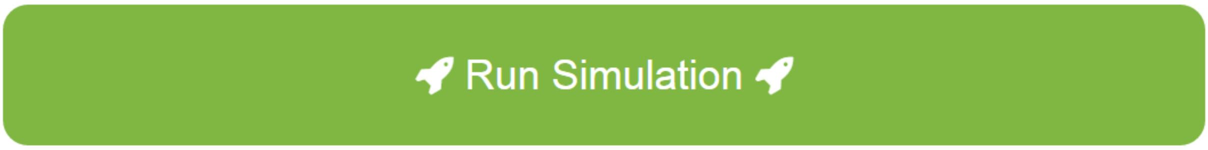


Step 3 - Choose the crop

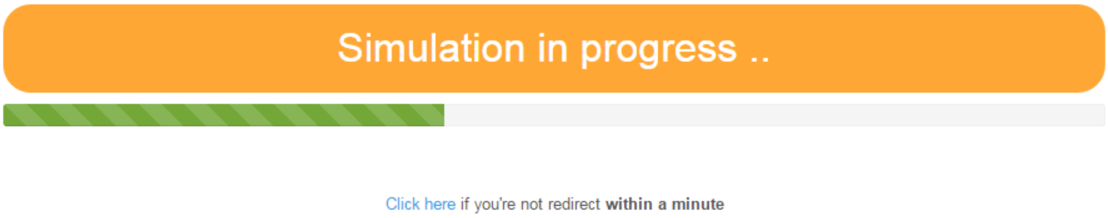
Crop: Wheat ✓

A green check will tell us that the crop has been recognized by our software.

We won't change *step 4*, so that we'll use the default date of planting, and we can run the simulation pressing the big green button on the bottom of the page



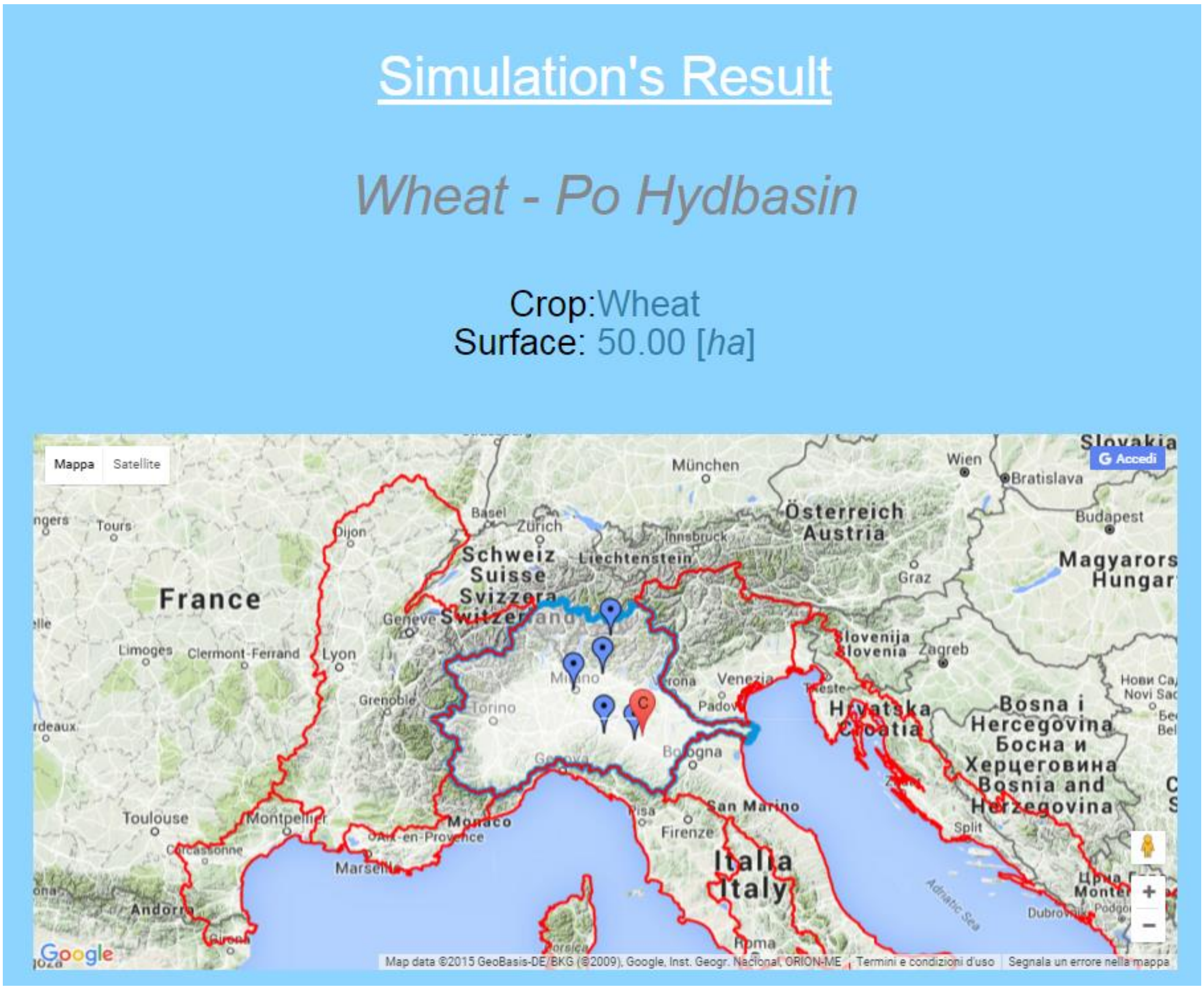
A window will let us know that the simulation is running and generally it will take a few seconds to be done. In case there's an error or the page won't change within a minute, just press the link on the bottom of the waiting page.



Simulation in progress ..

Click here if you're not redirect **within a minute**

Once it has finished, we'll be redirected to the result page. First we'll see a map like this:



Simulation's Result

Wheat - Po Hydbasin

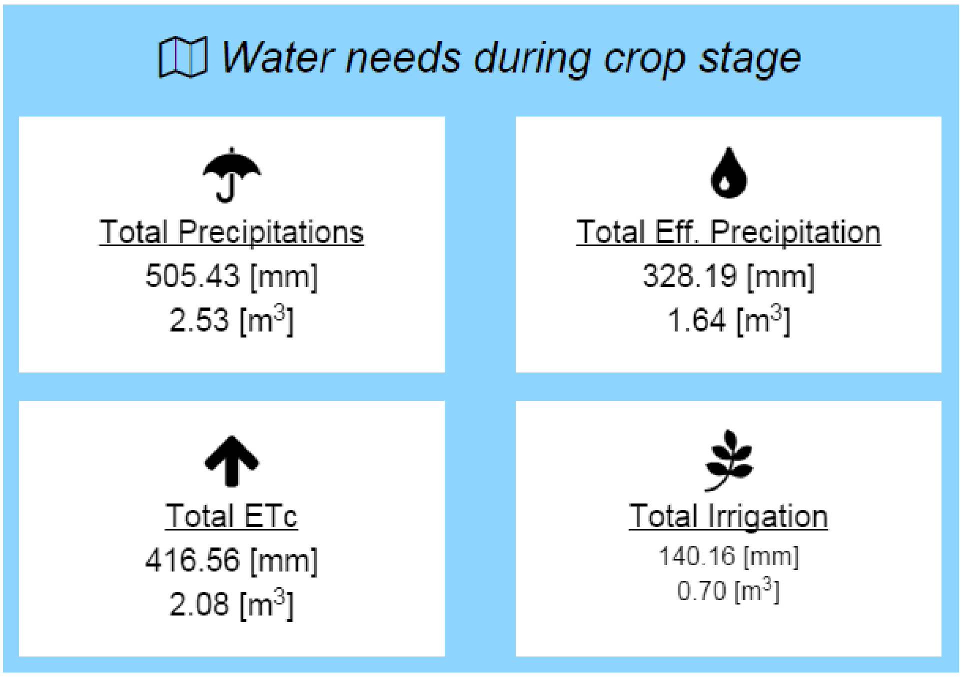
Crop:Wheat
Surface: 50.00 [*ha*]

On the top of the map we have the name of the simulation, with the surface and the name of the crop we've chosen. Under, we have the map with the point of simulation (the red marker) and the climate stations (blue markers). As we can see, there's not the station we've removed.

The hydrological basin is a little bit highlighted, but there are also shown some of the closest hydrobasins, that will give us an idea of how they're in that area.

Scrolling down, we first have a sum up of the results of simulation. All these results are in $\left[\frac{mm_{water}}{ha}\right]$ and also in $[m^3_{water}]$, thanks to the fact we've assigned a surface. These 4 boxes will tell us:

- The **total amount of precipitation**, that is the total amount of water that we expect it will rain on our surface during that period;
- The **total amount of effective precipitation**, that is the water that will be available for our plants, so we've excluded, for example, surface run off.
- The **total amount of ET_C**, so the evapotranspiration for our crop in our climate conditions. This is the total amount of water used by our crops in their all growing stage.
- The **total amount of irrigation**, so the water that the crop needs to avoid stress conditions


Water needs during crop stage

Total Precipitations
505.43 [mm]
2.53 [m^3]

Total Eff. Precipitation
328.19 [mm]
1.64 [m^3]

Total ETc
416.56 [mm]
2.08 [m^3]

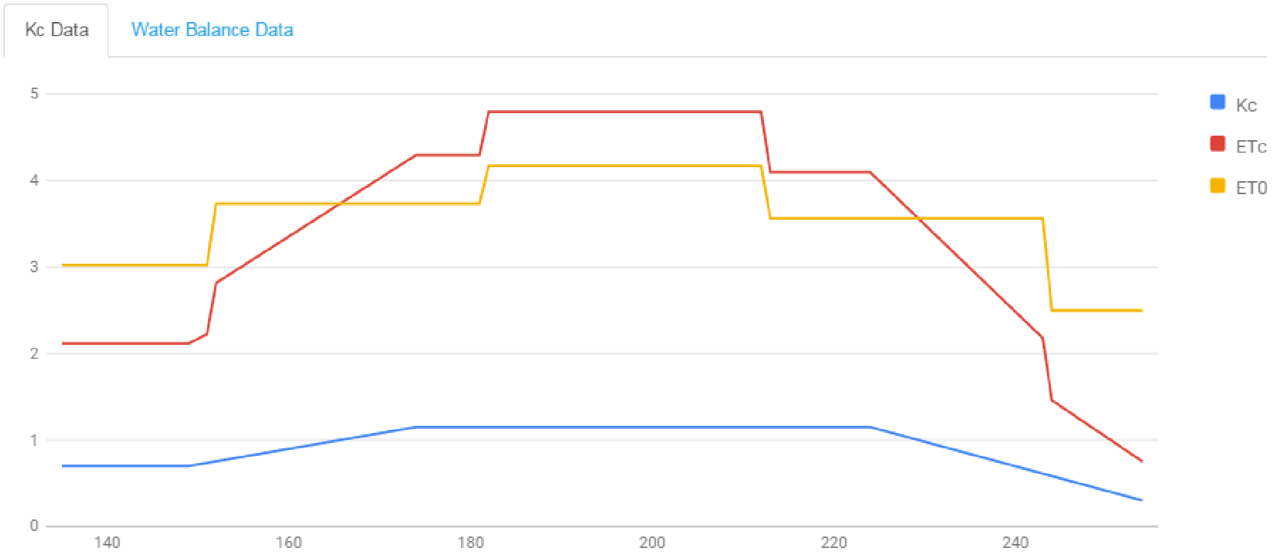
Total Irrigation
140.16 [mm]
0.70 [m^3]

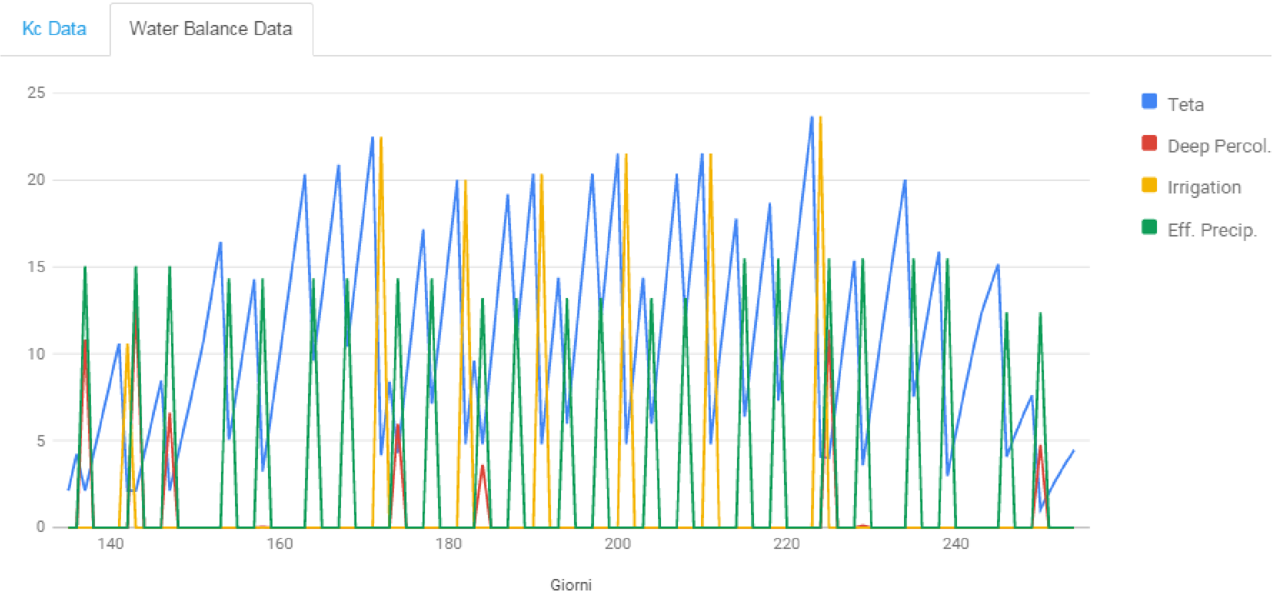
As mentioned also in the next section "*Future developments*", with the distribution of precipitation we've adopted (the *3 and *7 days of the month), sometimes we tend to underestimate irrigation need. This because, mainly in summer months, the rain is concentrate in certain days, and then we can have more than just 5 days without rain.

Finally, in the result page, we'll have under the button to download the results inside an Excel file, or we can just see them right under, in two different tabs. One tab, so call "*Kc Data*", shows us the values of K_C, ET_0 and ET_C. The other tab, called "*Water Balance Data*", will sum up all the values used for the water soil balance, that are:

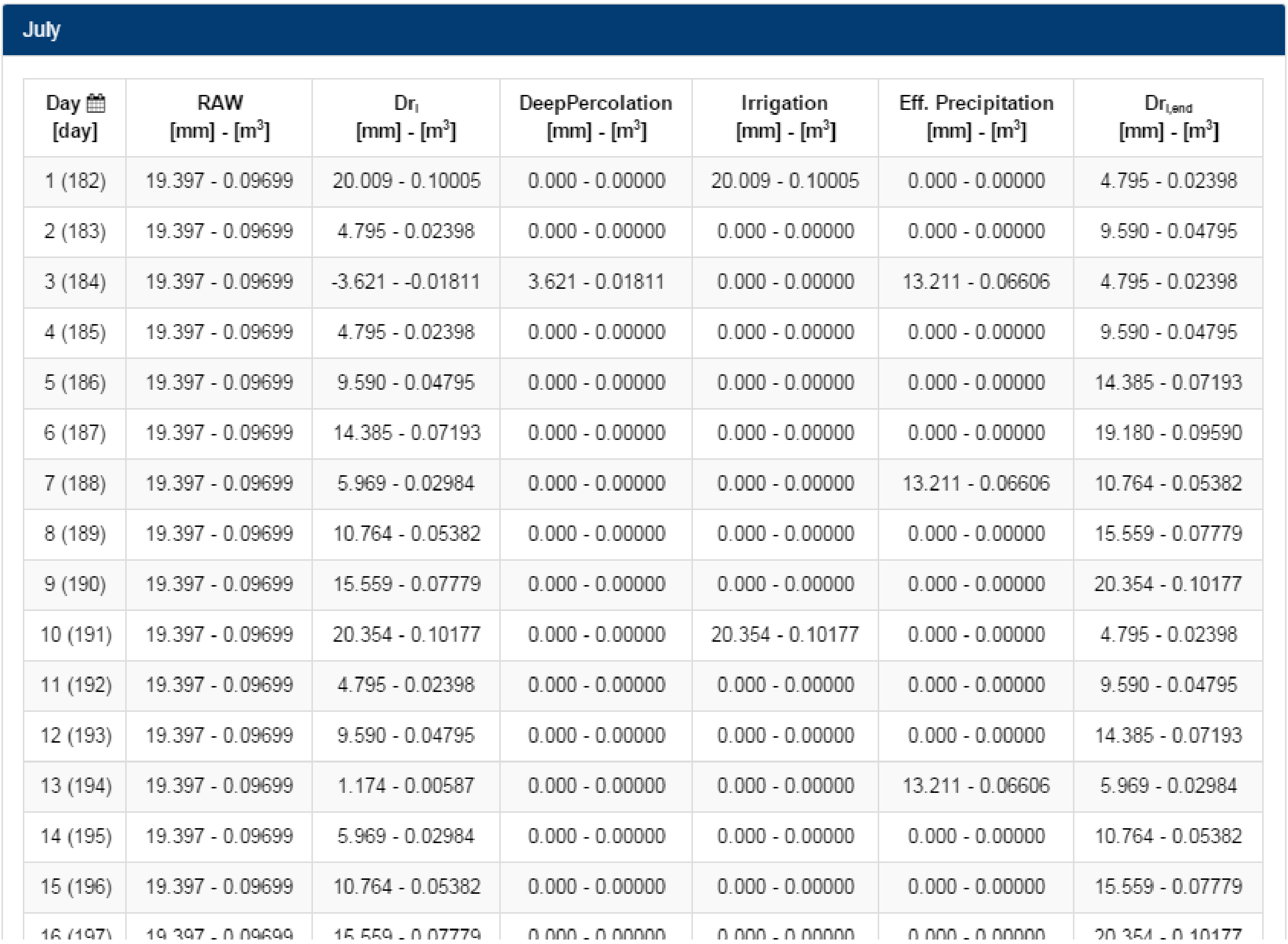
- Effective Precipitation, so the precipitation purified by run off and other water losses
- Irrigation, so the water given with human manufactures
- Deep percolation, so the water lost because the soil is not able to accumulate anymore
- Teta, so the level of water in the soil. It's expressed in millimeters and, as soon as it reaches a level that will create stress conditions to the crops, we activate irrigation and we bring that value close to 0.

The charts have, on the x axis, the days represented as the number of days since the beginning of the year. For our simulation, these are the two charts:





Under the charts, we can find the detailed numerical values. For example, we can see a part of the values for the month of July for what concern the *Water Balance Data*



July

Day [day]	RAW [mm] - [m^3]	Dr$_i$ [mm] - [m^3]	DeepPercolation [mm] - [m^3]	Irrigation [mm] - [m^3]	Eff. Precipitation [mm] - [m^3]	Dr$_{i,end}$ [mm] - [m^3]
1 (182)	19.397 - 0.09699	20.009 - 0.10005	0.000 - 0.00000	20.009 - 0.10005	0.000 - 0.00000	4.795 - 0.02398
2 (183)	19.397 - 0.09699	4.795 - 0.02398	0.000 - 0.00000	0.000 - 0.00000	0.000 - 0.00000	9.590 - 0.04795
3 (184)	19.397 - 0.09699	-3.621 - -0.01811	3.621 - 0.01811	0.000 - 0.00000	13.211 - 0.06606	4.795 - 0.02398
4 (185)	19.397 - 0.09699	4.795 - 0.02398	0.000 - 0.00000	0.000 - 0.00000	0.000 - 0.00000	9.590 - 0.04795
5 (186)	19.397 - 0.09699	9.590 - 0.04795	0.000 - 0.00000	0.000 - 0.00000	0.000 - 0.00000	14.385 - 0.07193
6 (187)	19.397 - 0.09699	14.385 - 0.07193	0.000 - 0.00000	0.000 - 0.00000	0.000 - 0.00000	19.180 - 0.09590
7 (188)	19.397 - 0.09699	5.969 - 0.02984	0.000 - 0.00000	0.000 - 0.00000	13.211 - 0.06606	10.764 - 0.05382
8 (189)	19.397 - 0.09699	10.764 - 0.05382	0.000 - 0.00000	0.000 - 0.00000	0.000 - 0.00000	15.559 - 0.07779
9 (190)	19.397 - 0.09699	15.559 - 0.07779	0.000 - 0.00000	0.000 - 0.00000	0.000 - 0.00000	20.354 - 0.10177
10 (191)	19.397 - 0.09699	20.354 - 0.10177	0.000 - 0.00000	20.354 - 0.10177	0.000 - 0.00000	4.795 - 0.02398
11 (192)	19.397 - 0.09699	4.795 - 0.02398	0.000 - 0.00000	0.000 - 0.00000	0.000 - 0.00000	9.590 - 0.04795
12 (193)	19.397 - 0.09699	9.590 - 0.04795	0.000 - 0.00000	0.000 - 0.00000	0.000 - 0.00000	14.385 - 0.07193
13 (194)	19.397 - 0.09699	1.174 - 0.00587	0.000 - 0.00000	0.000 - 0.00000	13.211 - 0.06606	5.969 - 0.02984
14 (195)	19.397 - 0.09699	5.969 - 0.02984	0.000 - 0.00000	0.000 - 0.00000	0.000 - 0.00000	10.764 - 0.05382
15 (196)	19.397 - 0.09699	10.764 - 0.05382	0.000 - 0.00000	0.000 - 0.00000	0.000 - 0.00000	15.559 - 0.07779
16 (197)	19.397 - 0.09699	15.559 - 0.07779	0.000 - 0.00000	0.000 - 0.00000	0.000 - 0.00000	20.354 - 0.10177

The columns have the following meanings:

- **Day** is the day of the month. The number inside the brackets is the number of the day since the beginning of the year.
- **RAW** is the value in millimeters where, at that stage of the crop, we start having crop under stress if the water is not present
- $\boldsymbol{Dr_i}$ is the value, in millimeters, of the actual level of water at the beginning of the day. $20\ [mm]$ means that we have the water level $2\ [cm]$ under the surface
- **DeepPercolation** is how much water we lose, and we have only during rainy days
- **Irrigation** is the amount of water given by irrigation and it's present only when we have crops under stress conditions
- **Effective Precipitation** is the precipitation, purify by unused water such as run off, and it's given to the system only in *3 and *7 days
- $\boldsymbol{Dr_{i,end}}$ is the level of the water, in millimeters, at the end of the day. This is what, on the charts, is present as **Teta**

All these values are also available into the excel file, where we can also find the same charts already made, ready to be used.

Future developments

This tool is open to a lot of further implementations, some of those I've tried to sum up in this brief paragraph.

First of all, we can let the user decide the precipitation days. This means, the user will be able to maybe concentrate the rain in just 2 or 3 days inside the month, let the irrigation pay a bigger role. This will make the irrigation values closer to the reality values.

For the rice, we should add a criteria to evaluate better the water need, since now the value is really underestimate.

The climate stations can be added by the user, with its custom data. This will help also to raise the number of values, since the new stations can be shared through the users.

Finally, a user can edit their previous simulations, without making every time a new simulation. That will result in a less time waste.

6. Minimize resources maximizing production

Now that we have spent a lot of time talking about how to evaluate the water consumption, let's try to find out a criteria to see if there's a way to reduce the actual water consumption in the world.

Nowadays, a big problem is the massive utilize of water and lands as an answer to the growing request of food. What we want to find out is, utilizing the database made in the previous chapter and adding new existing information about production, if there's a way to maximize the production reducing the water used and not using a single hectare of land more.

This study has been made looking at 3 factors:

- **kg**: the kilograms of total production
- **kcal**: the kilocalories made out of production
- g_{prot}: the grams of protein produced

Since we cannot include all the crops, we decided to take into consideration the 14 main crops in the world, that represent the 90% of the total actual production. These are the following:

- cassava
- groundnut
- maize
- millet
- oil palm
- potato
- rape
- rice
- sorghum
- soybean
- sugar beet
- sugarcane
- sunflower

- wheat

Gaez FAO: our data source

First of all, we need a way to integrate our database with current production data. To be consistent, we decided to use the data taken from the GAEZ FAO project website (*Global Agro-Ecological Zones*), public accessible at

http://gaez.fao.org/

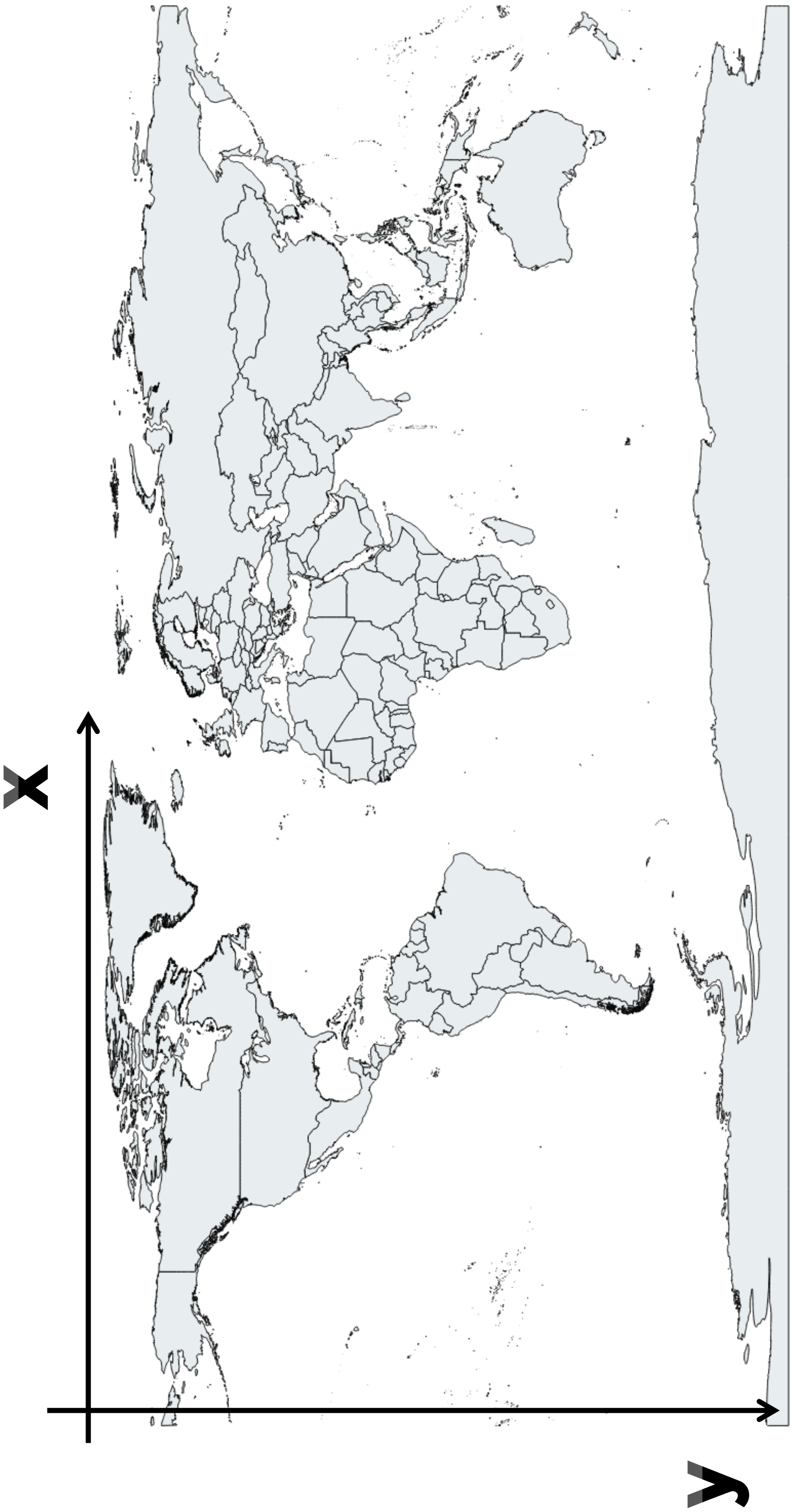
The data have been downloaded as a .tiff file, that, thanks to the *gdal* tool, a Linux open source tool used by a lot of GIS software, such as QGIS, we were able to transform them into .xyz file. This kind of files are like CSV files with space separation instead then comma separation, plus they allow just 3 columns, and not infinite as CSV. So, the first column of the file will be our x coordinate, the second our y coordinate and the 3rd one the value we want to read.

This obliged us to divide the surface into points, since we'll have point data and not spatial data. We decided to keep the resolution of .tiff file, in order not to loose information, so we decided to have a value every $0.08\overline{3}°$, both for latitude and longitude. For most of the points on earth, this means a distance of around $10\ [km]$.

Finally, a consideration about fertilizers. We've chose to use an intermediate input of fertilizer, avoiding the high input. This will give us two benefits:

- reduce the impact of fertilizers on nature;
- gives us the possibility to produce more, compared to our calculations, thanks to the possibility to increase fertilizer input.

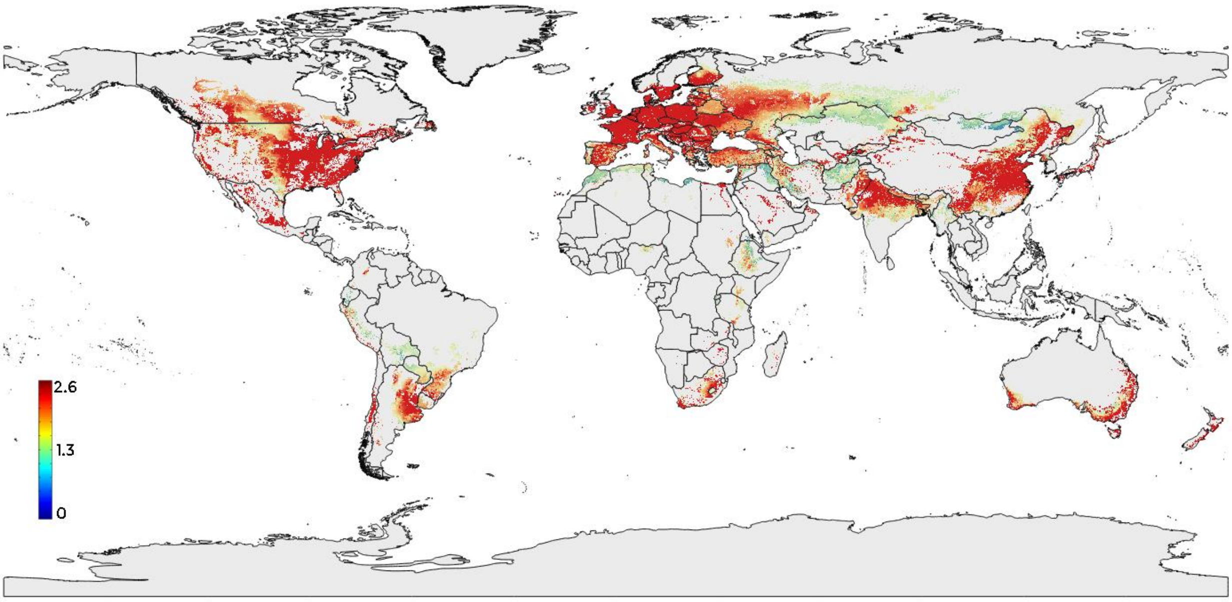
Since all the maps has the same number of points with the same distance, we created a simple x-y coordinate system as represented in the following image. This allowed us to save space in our database, together with increasing the speed of searches. Ad-hoc created functions allowed us to pass easily from latitude and longitude values, to x and y values. Since the map is divided into pixels, so square shapes, we decided to put our reference point in the middle of the square, obtaining the following map. The dimensions are 4320 x 2160 cells, and the coordinates in *x* go from 0 to 4319, and in *y* from 0 to 2159..


x
y

Step 1: find yield of crops

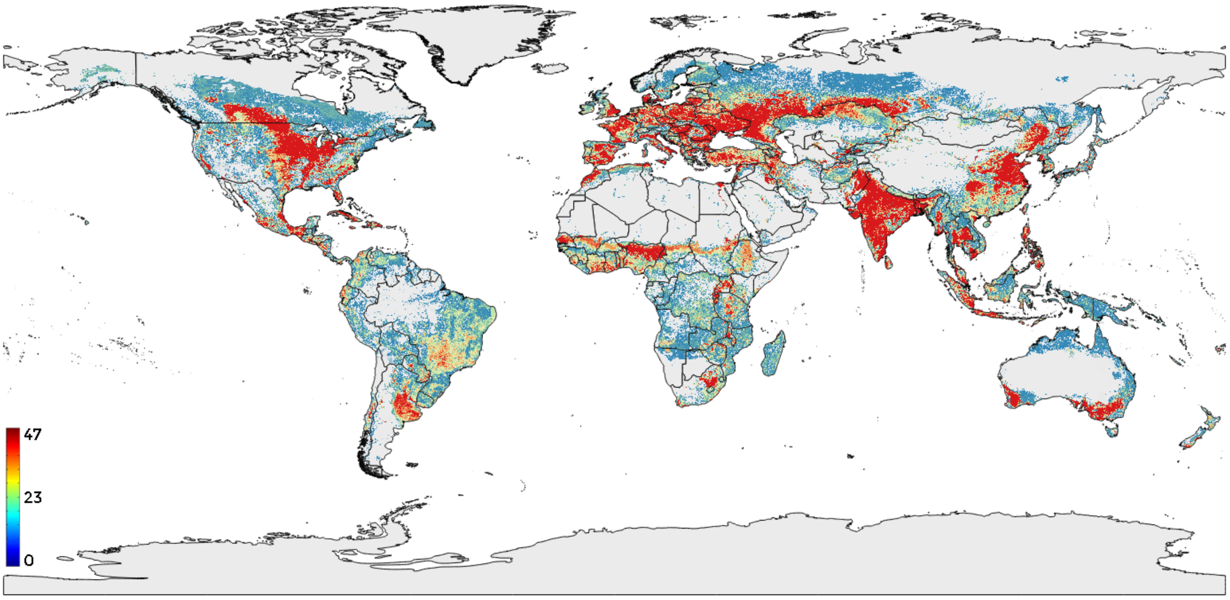
To start, we need to now the actual yield of the crops currently in the world. That means, the value of how much we produce a year in terms of $[kg]$ in a defined area. That leads, typically, into values with unit of measure of $\left[\frac{kg}{ha \cdot year}\right]$. All our calculations are done base on year values, so we can omit the $year$ unit in our equations.

We've downloaded the actual yield maps for the 14 crops of study. One actual yield map looks like the following image, that is the wheat actual production in $\left[\frac{tons}{ha}\right]$.



Since we want to increase production and decrease water footprint, maintaining the same land use, that means we have to redistribute the crops, so maybe cultivate a crop where it's not cultivated today. Of course, we have to make sure we won't have a crop where it's impossible for that crop to grow, for example we cannot have oil palm in Canada.

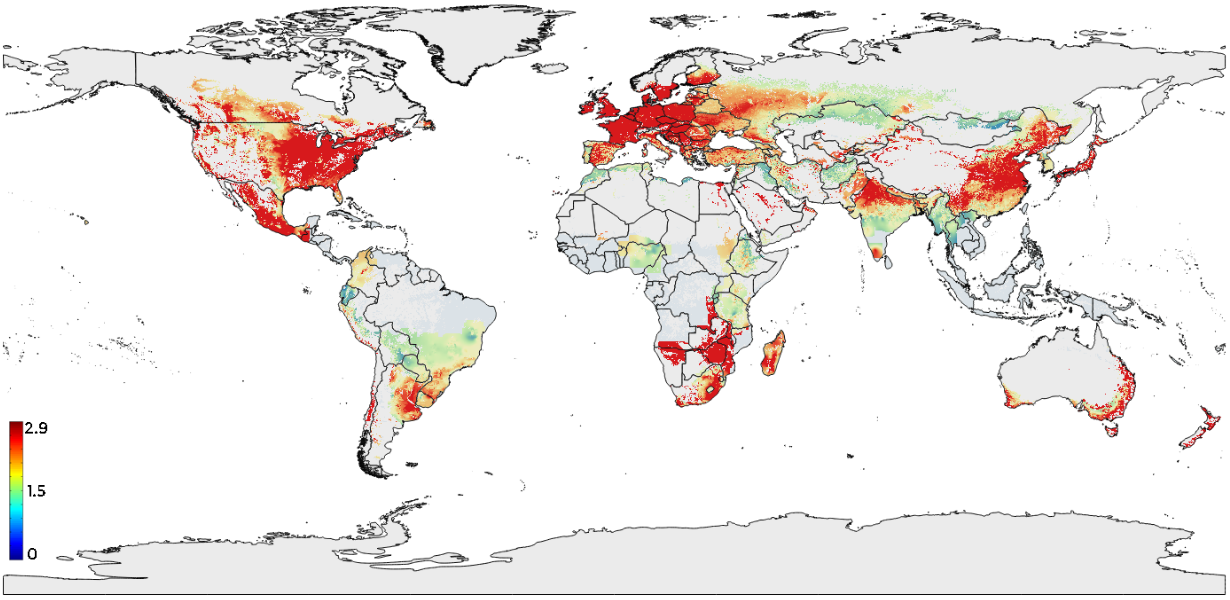
Starting from these 14 maps, we then recreate the *cut maps*, and that's what we've done: we've taken the current cultivated land map, that represent the current cultivated lands in the world. We can see the map in the following image: each pixel has a percentage that indicates how much land is cultivated in that pixel. A pixel corresponds to a surface of around $100\ [km^2]$ or $10000\ [ha]$.



Utilizing a spline algorithm inside *ArcGIS*, we made 14 maps that in all the current cultivated lands points had a value of yield. Of course, that means we will have oil palm in Canada, since in Canada we have cultivated lands. The next step has been, so, to make sure we have values in each point not further than a certain radius. We decide to choose a radius of $250\ [km]$ or $25\ [pixels]$.

The number 25 has been choosing after a few simulations of different radius based on soil suitability. Choosing a bigger radius, we were going to have a big uncertainty in soil suitability. As long as the radius was under $25\ [pixels]$, the suitability had a variation in that radius of 14% as average.

The result of the cutting are 14 maps that allow us to have yield values where FAO was not able to give us. Just for an example, the following map is the *actual yield cut map* for the wheat.



Before proceed with step 2, we made also another consideration: values of actual yield depends also from countries. Developed countries with high technology reflects into higher yield values. So, we applied again the spline algorithm, this time considering the borders of the countries. This will results in edges sometimes, with remarkable differences between countries, such as those between Israel and Palestine. In this way, cutting the maps always with the criteria of $25\ [pixels]$ radius, we had other 14 maps with slightly different values. Just for comparison, the next map is the *actual yield cut countries map* for wheat.

Step 2: optimize kg, kcal and gr_{prot}

We haven't talked yet about water, and won't do till next step. Now we have the production all over the world in terms of $\left[\frac{kg}{ha}\right]$. With these values we can start making a simple improvement: for each cell of the map (each pixel), we see which is the crop that will produce the biggest amount in terms of $[kg]$.

As we can understand, we're making a big assumption: **when we replace crops in a cell, all the cultivated land in that cell will produce that crop**. This is an ideal transformation, to make calculation easier and give a global value of what we can reach as goal.

To obtain the values into $[kcal]$ or $[gr_{prot}]$ we need conversion values. Each value of $[kg]$, so each pixel in the maps, we'll transform into $[kcal]$ or $[gr_{prot}]$. To do that, we need transformation coefficients, that we can find in the following table:

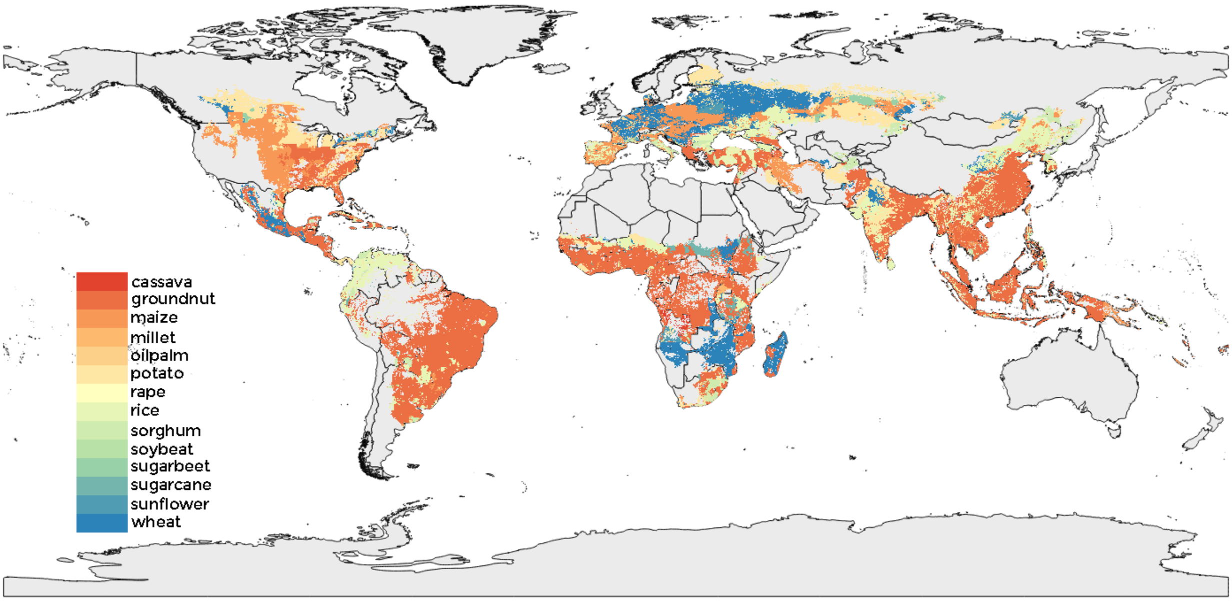
	$\left[\frac{\boldsymbol{kcal}}{\boldsymbol{kg}}\right]$	$\left[\frac{\boldsymbol{gr_{prot}}}{\boldsymbol{kg}}\right]$
cassava	996,07	6,55
rice	3695,42	69,50
groundnut	5373,83	233,64
maize	3008,30	72,82
millet	2941,18	79,52
oilpalm	1592,79	0,31
potato	588,24	14,12
rape	8780,49	0,24
sorghum	3072,35	90,19
soybean	3465,35	2,48
sugarbeet	50,00	5,00
sugarcane	319,74	1,60
sunflower	3448,28	103,45
wheat	2930,81	88,54

The criteria to figure out the map with the best production it's very simple: for each point in all the 14 maps, we figure out which is the crop with the highest production in that cell. This crop, and its value of production, will be the value that we'll have in the final cell. So, at this stage, we have:

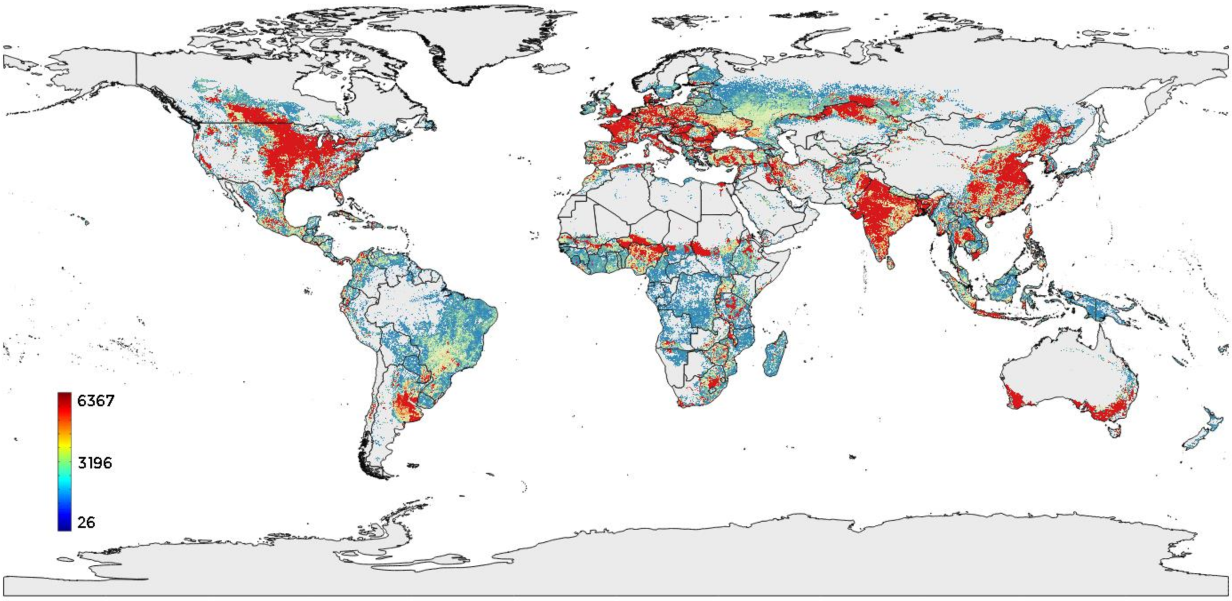
- one map for maximizing $[kg]$ scenario, that shows us the crop distribution in our cultivated lands, as well as another map that will show us the value of production;
- other two maps, but for the $[kcal]$ scenario, one for the crop distribution and another one for the production
- same for $[gr_{prot}]$, with one map for crops and one map for values

The results are in the maps inside Appendix 2. Here we'll report just one map that will show the results for what concern $[gr_{prot}]$. Trying to maximize this value, we found out a crop distribution. For that crop, we've printed the values of $[kg]$, $[kcal]$ and $[gr_{prot}]$.

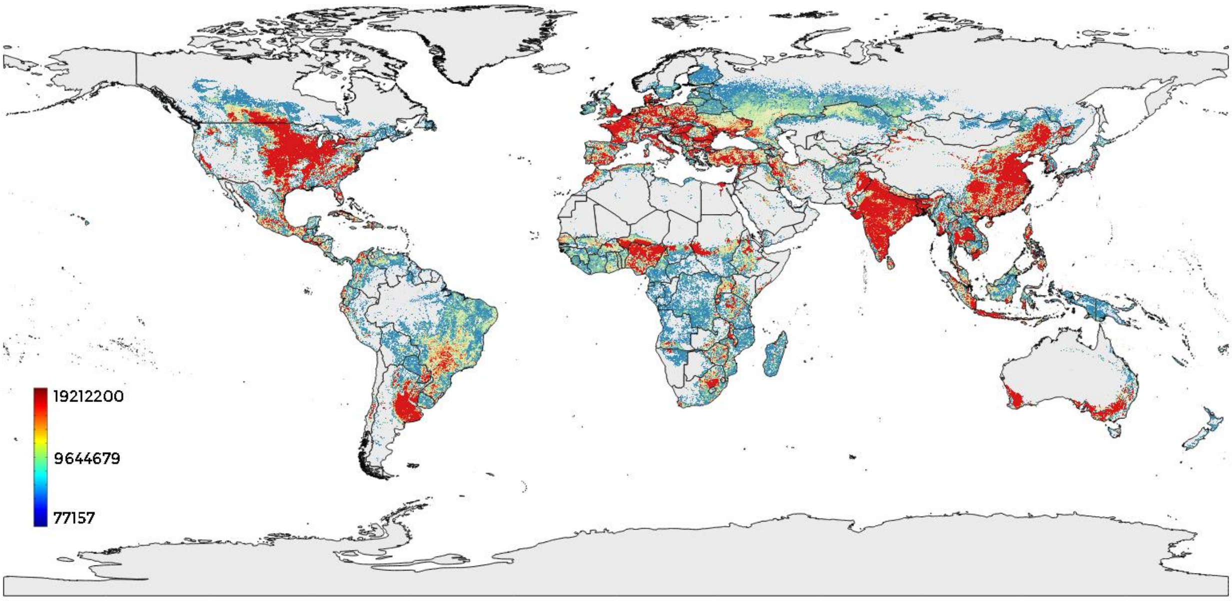
Crop distribution



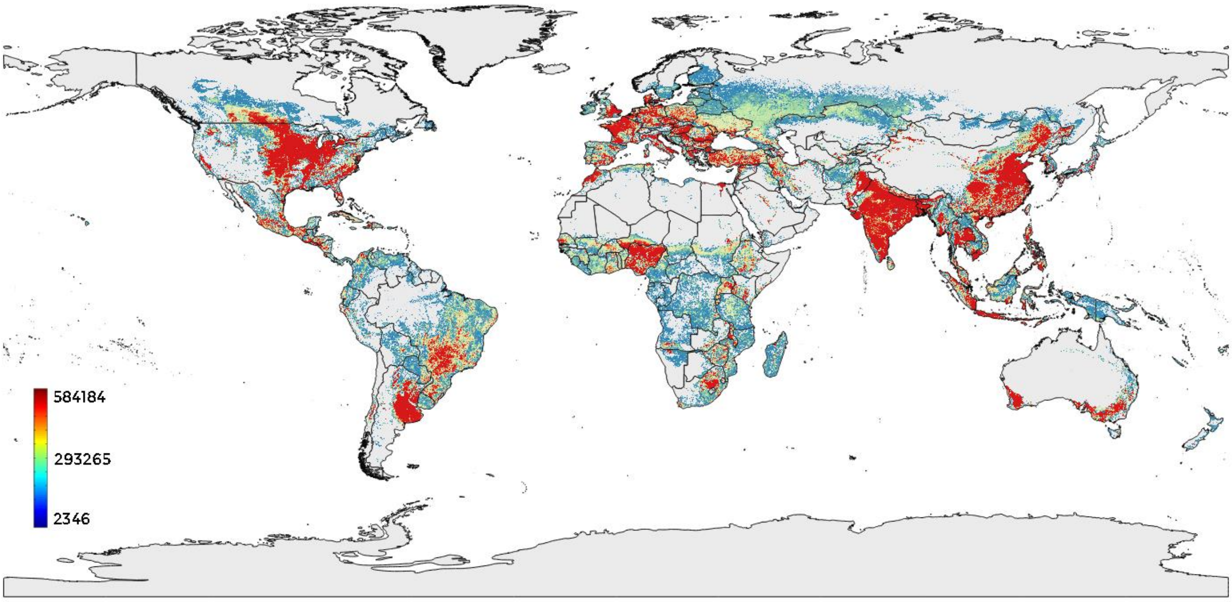
Tons produced



Gcal $[10^9\ Kcal]$ *producted*

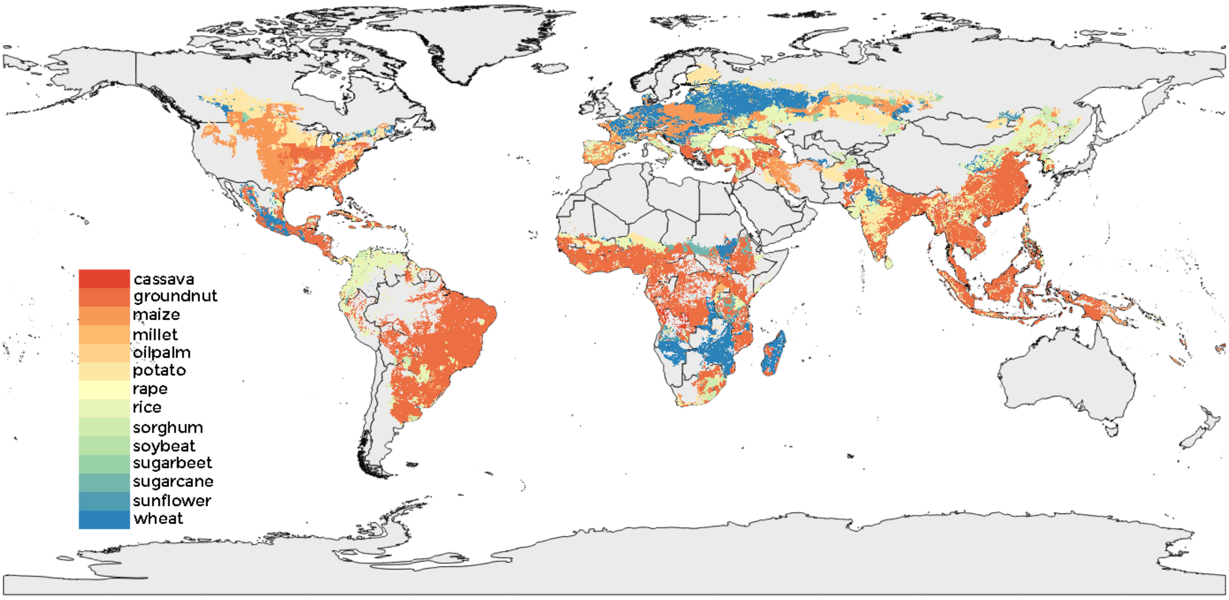


Kg of proteins produced

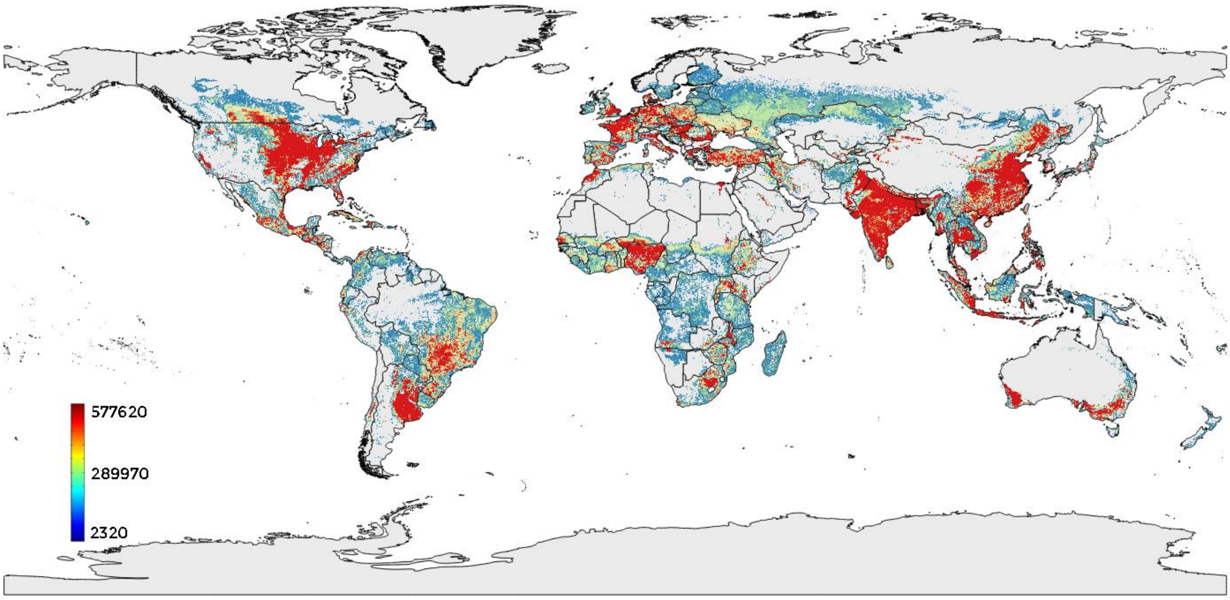


We did also the same calculations utilizing the yield obtained taking care of country borders. The results are not so much different, and just for purpose of comparison we'll report the map of the crop distribution, and the values in $[kg_{prot}]$ for the distribution that maximize the $[gr_{prot}]$ value.

Crop distribution



Kg of proteins produced



Step 3: Minimize Water, maximize production

Of course just looking at kg, $kcal$ or gr_{prot} on their own has a little meaning, cause we haven't seen yet the amount of water saved. Maybe we're able to produce more, but we have to make sure that we won't consume more water, but if it's possible, we need to consume less.

First of all, we needed to import the water footprint values given by M. M. Mekonnen and A. Y. Hoekstra in each point of the world (2011). A map with values in $\left[\frac{mm}{ha}\right]$. From this values, we can evaluate the real water footprint, based on the crop cultivated and the surface of cultivated land we have in each cell. Like we did for the yield, we'll extend these values based on radius of 25 cells, so that we can have values in places where FAO is not reporting them.

Multiplying those values for the surface of each cell, we can get the amount of water used by that crop inside that cell.

To find out the crop distribution we're looking for, we'll implement the following algorithm:

- Move cell by cell in our whole map

- For each cell, we'll go from the crop that consume less water to the one that consume more water
- We'll start from the first crop of that list: if the production is equal or more than the actual production in that cell, we keep that crop, otherwise we'll go to the next crop, and so on, until we meet a crop that produce equal or more.

The "production" parameters we take under consideration will be once kg, once $kcal$ and once gr_{prot}. So we'll have 3 different crops distribution:

- one that minimize the water footprint making sure that we're not producing less (in terms of kg) compared to actual production
- one that minimize the water footprint making sure that we're not producing less (in terms of $kcal$) compared to actual production
- one that minimize the water footprint making sure that we're not producing less (in terms of gr_{prot}) compared to actual production

And these are 3 distributions that take under consideration water, and not just production. All these maps can be found inside the Appendix 2. As you can see, for each distribution we've represented all the possible values, so the kg, $kcal$ and gr_{prot} produced, and also the m^3_{water} used for growing those crops.

To see how the scenario improved, compared to the actual situation, we can have a look at the following sum up tables.

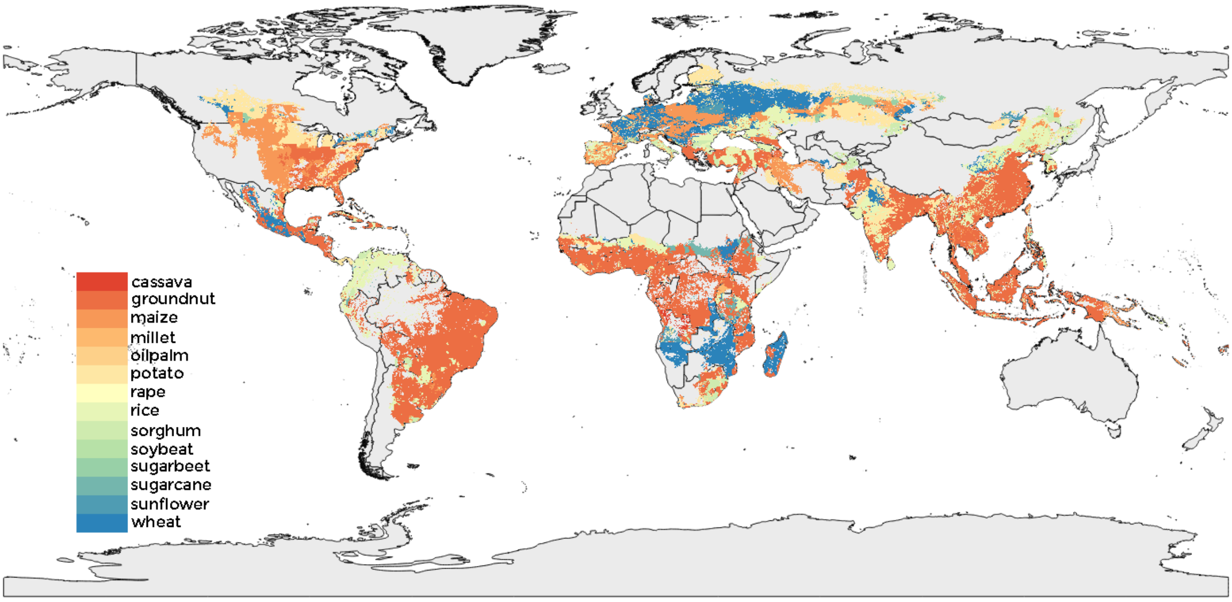
Let's start with the situation derived from the actual yield map not considering the borders between countries.

	kg	$kcal$	gr_{prot}	$H_2O - kg$	$H_2O - kcal$	$H_2O - gr_{prot}$
kg	$811\ \%$	$385\ \%$	$31\ \%$	$591\ \%$	$59\ \%$	$-6\ \%$
$kcal$	$18\ \%$	$181\ \%$	$22\ \%$	$4\ \%$	$107\ \%$	$7\ \%$
gr_{prot}	$-32\ \%$	$17\ \%$	$104\ \%$	$-7\ \%$	$29\ \%$	$89\ \%$
m^3_{water}	$78\ \%$	$48\ \%$	$-20\ \%$	$46\ \%$	$-3\ \%$	$-27\ \%$

The first consideration that we do is that when we try to maximize the production in term of kg and $kcal$, we will use really a lot of water more.

Then, it's not possible to minimize water making sure we'll produce more in terms of kg. A person would expect here to find $0\ \%$for what concern water used, cause we said that our algorithm will take only the values that will consume less water with equal production. The reason why we have an increase of water use is due to the fact that in each cell we're replacing the same crop everywhere, and we're not considering more than one crop in the same cell.

The criteria that creates a better situation is the gr_{prot} criteria, where we increase the production of everything, decreasing water use. The crop distribution is the following:



Compared to the other situations, it's also a distribution that guarantees a good homogeneity of these crops in the world. In fact, if we look at distribution just based on kg criteria, we'll have an enormous amount of sugarcane and sugar beet all over the world.

Step 5: Considering soybean

In the previous calculations, we treated soybean as one of the other crops, and in our replacements was almost always removed. In fact we have that, compared to actual situation, the 99% of land cultivated with soybean has been removed.

If for our previous algorithm that was acceptable, it's not acceptable from a real point of view, since soybean is able to obtain its own nitrogen (N) through the process of N-fixation. In other words, it's used, alternated with other crops, to let the soil rest for one year and naturally recover the nutrients for the next season.

Mainly for this reason, we decided to maintain the crop distributions we found out in the previous steps, but if a cell has a part of the cultivated land actually used by soybean, we'll maintain that value. This will for sure results into a reduction of the previous values, and the values, always in percentage, are reported in the following table

	kg	$kcal$	gr_{prot}	$H_2O - kg$	$H_2O - kcal$	$H_2O - gr_{prot}$
kg	$710\ \%$	$329\ \%$	$19\ \%$	$507\ \%$	$37\ \%$	$-16\ \%$
$kcal$	$13\ \%$	$152\ \%$	$14\ \%$	$-1\ \%$	$91\ \%$	$1\ \%$
gr_{prot}	$-41\ \%$	$1\ \%$	$78\ \%$	$-17\ \%$	$16\ \%$	$66\ \%$
m^3_{water}	$68\ \%$	$40\ \%$	$-20\ \%$	$39\ \%$	$-4\ \%$	$-26\ \%$

Even though we have a few values reduced, still the situation that maximize gr_{prot} is the best one, since it allows an increase of production in all the 3 values that we want to maximize, reducing the water utilize and, most of all, not using a single hectare of land more.

The results that are show in the maps inside Appendix 3, can also be seen divided by countries. We ordered them alphabetically, based on the name of the country. In the next page there's an extract of that appendix, that shows the 10 most productive states in the world, in our redistributed scenario. The scenario we chose is the crop distribution maximizing gr_{prot} and we reordered them based on the water footprint (from the country that will use more water to the one that uses less water).

	Cassava	Groundnut	Maize	Millet	Oil Palm	Potato	Rape
China	0	53510013	855445	266	0	826106	0
United States of America	0	13121213	19265001	0	0	7360698	0
India	9451	15829745	11640	0	0	9663703	0
Brazil	0	9320939	169915	0	0	106860	0
Australia	0	0	131601	0	0	5650659	0
Nigeria	3	7407898	0	0	0	12165	0
Argentina	0	6312460	317181	0	0	37204	0
Indonesia	158	6418796	8716	0	0	2461	0
France	0	0	1852588	0	0	856221	0
Russian Federation	0	355968	1073406	0	0	254934	0

	Rice	Sorghum	Soybean	Sugar beet	Sugarcane	Sunflower	Wheat	Total
China	6908785	1224316	0	578	0	1343	929391	64256243
United States of America	17332	787	0	3008	0	132	12491	39780662
India	5518071	0	0	0	0	0	1684472	32717082
Brazil	171532	0	0	0	2	0	0	9769248
Australia	3774658	27488	0	0	239	46	3634	9588325
Nigeria	1254676	0	0	0	4	0	64087	8738833
Argentina	198723	720323	0	0	0	17	3756	7589664
Indonesia	1035129	0	0	0	15	0	0	7465275
France	0	269706	0	55184	0	0	3230608	6264307
Russian Federation	1326164	36762	0	298484	0	37407	2483992	5867117

Future possible improvements

What we've calculated are not all the possible solutions we can find. First of all, we can remove the assumption that all the cell is planted with the same crop, but we can even define different percentages of cultivated land for different crops.

Another aspect is to consider not only the border between countries, but also the hydrological basins in the world, when we recreate the actual yield in surrounding areas.

We then should be able to consider not just the difference in gross price, but the difference in net price. Cause maybe a farmer will earn less producing a new crop, but if the net price will increase, that means the farmer will earn more. Of course, this will require a better analysis of a lot of different factors, such as:

- the techniques of production used for each crop in each part of the world;
- the cost for starting a new production of a crop, rather then keep staying with the same crop

For what concern the techniques, we should also be able to evaluate the difference in production improving the techniques used in less developed areas, reaching a better water efficiency.

7. Close the yield gap: 15 countries study

Case study

To show how this study can be connected to improve previous studies, we run our code in a multiple calculation to complete the study of Francesco Laio, published in December 2015 after a collaboration between the Politecnico of Torino and the University of Virginia (UVA).

The first part of that study found the 15 major crops, covering the 73% of planet's agricultural area, and found also a number of countries that are suffering of irrigation-limited yield. That is, we are able to produce more in those areas, just increasing the water given through irrigation (blue water).

The crops chosen for the studies are the following:

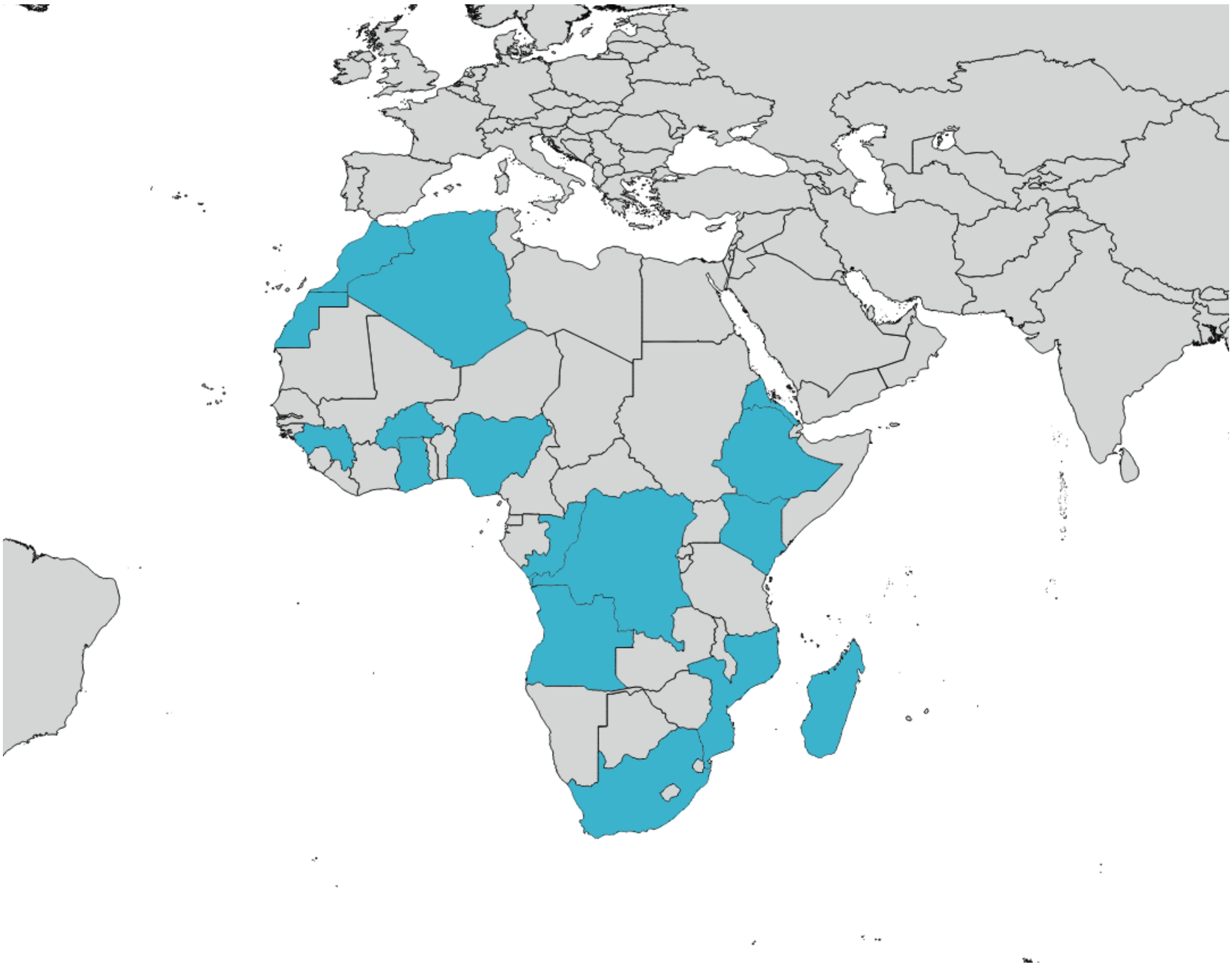
- barley
- groundnut
- maize
- millet
- oil palm
- potato
- rape
- rice
- rye
- sorghum
- soybean
- sugar beet
- sugarcane
- sunflower
- wheat

And the countries are the following:

- Algeria
- Angola

- Burkina Faso
- Congo
- Democratic Republic of Congo
- Eritrea
- Ethiopia
- Ghana
- Guinea
- Kenya
- Madagascar
- Morocco
- Mozambique
- Nigeria
- South Africa

The graphical representation of those 15 countries, all inside the continent of Africa, is given in the following map.



Each of these countries don't produce all the above listed crops, but just a part. The following table sums up the crops that are planted in each country that we'll take under consideration.

	B	*G*	*M*	*ML*	*O*	*P*	*RA*	*RY*	*SG*	*SY*	*SC*	*SF*	*W*
Algeria	X	X				X	X		X				X
Angola		X	X	X		X					X	X	
Burkina Faso		X	X	X		X			X	X			
Congo		X			X	X					X		
Dem. Rep. Congo		X	X	X	X	X			X	X	X		
Eritrea	X	X		X		X			X				
Ethiopia	X	X	X	X		X			X				X
Ghana		X	X	X	X				X		X		
Guinea		X		X	X				X				
Kenya		X	X	X		X			X		X	X	
Madagascar		X	X			X			X		X		
Morocco	X	X	X	X		X	X	X	X	X	X	X	X
Mozambique		X	X	X		X			X		X	X	
Nigeria		X	X	X		X			X	X	X		
South Africa	X	X	X	X				X	X	X	X	X	X

For the crops, these are the meaning of the columns:

- B-Barley,
- G-Groundnut,
- M-Maize,
- ML-Millet,
- O-Oil palm,
- P-Potato,
- RA-Rapeseed,
- RY-Rye,
- SO-Sorghum,
- SY-Soybean,
- SC-Sugarcane,
- SF-Sunflower,
- W-Wheat.

Results

First of all we decided to evaluate the water need just in the cells that are considered cultivated, according to FAO data downloaded for the minimizing water part of our paper (chapter 6).

For each of these cells, we obtained the values in terms of $\left[\frac{mm}{ha\cdot yr}\right]$, so that they can be easily readapted for different land surfaces.

We have to remember that those values are inside optimum conditions, but they give us a good view of the really values in real life situations. The numerical values can be found, divided for each state, in the following tables:

	Barley	*Groundnut*	*Maize*	*Millet*	*Oil Palm*	*Potato*	*Rapeseed*
Algeria	122,01	281,40				296,82	250,62
Angola		225,55	243,11	160,54		220,96	
Burkina Faso		141,28	61,54	20,52		158,61	
Congo		97,30			17,21	50,88	
Dem. Rep. Congo		78,62	0,00	0,00	23,14	54,58	
Eritrea	156,48	201,45		62,29		148,54	
Ethiopia	198,75	206,78	114,66	37,36		268,44	
Ghana		265,30	84,90	23,79	457,94		
Guinea		501,23		32,24	1080,64		
Kenya		100,83	124,56	49,07		103,01	
Madagascar		50,81	151,93			41,64	
Morocco	171,56	264,05	323,61	257,60		296,37	235,24
Mozambique		139,24	211,94	115,30		129,01	
Nigeria		432,46	110,53	19,98		503,86	
South Africa	143,90	157,18	122,95	63,96		0,00	

	Rye	*Sorghum*	*Soybean*	*Sugar Cane*	*Sunflower*	*Wheat*
Algeria		182,04				135,74
Angola				583,03	88,70	
Burkina Faso		31,92	103,67			
Congo				101,93		
Dem. Rep. Congo		0,00	53,55	174,96		
Eritrea		85,32				
Ethiopia		61,17				312,30
Ghana		26,65		432,10		
Guinea		57,07				
Kenya		53,31		418,66	64,13	
Madagascar		101,13		422,15		
Morocco	248,25	188,56	314,48	627,05	69,62	202,69
Mozambique		140,97		559,20	17,86	
Nigeria		31,15	277,52	883,69		
South Africa	151,37	80,08	140,49	517,00	122,97	236,23

For what concern some crops, such as Barley or Maize, there's not a big difference between the values, except for a couple of countries. But for the most there's a big difference between the different countries. If we see, Oil Palm, we see how in Congo we don't need a lot of water to close the yield gap, but in Guinea we're very far from optimum production if we don't use a big amount of water to allow oil palm growing.

Of course these values are an average on the cultivated lands of those countries, and can vary significantly from field to field in the same country. Just consider that South Africa has a total surface of 1219090 $[km^2]$, and to get the exact value in each point we can use the online tool of WaterGrab.Zone.

8. Conclusions

Evaluate water need to grow a crop in a certain part of the world, under our hypothesis, is now an open source tool for everyone, easy to use and well customizable. Of course, as we've indicated in the appropriate sections, there are other thousands of improvements possible. The orientation of these improvements should be first to make the calculation more accurate. In this section, the software has already been studied to work with a lot of possible customization such as:

- soil type change compared to the actual one
- each user can upload its own climate stations, in order to
    - increase the number of information
    - use more accurate information
    - simulate possible future changes in climate conditions
- the parameters for each crop can be user added

The aim of this total personalization is to allow each user to create a unique database where information can be easily shared, without having a big dispersion between the different websites of the different university or organizations.

We also demonstrated how our planet can feed more than the actual population. Of course we assumed a scenario that, due to the logic of capitalism, it's hard to keep alive. As the results show, there's still a big margin of variation so that we can still imagine that situation that, beside the increase of production, guarantee a reduction in water footprint.

In addition to our calculations, we checked the prices of the new crop distribution compared to the actual distribution. The prices are divided for country, since they strongly depend from the level of technologies used and the cost of transportation (level of infrastructures, price of gas, etc). Comparing the production obtain with our calculation show that in the 74% of the places there's no a reduction in terms of total income. A more accurate study can be conducted taking under consideration not just the total income, but the net income. This extra consideration needs a deep study for all the countries in the world, cause the cost of production vary from area to area. Sometimes it has strong differences also inside the same country.

What's sure is that to make a more realistic situation, we need to include in each cell at least 4 crops. The criteria will be the same used, that is minimizing the water footprint, making sure that the total production it is not increasing. Of course, with 4 crops we'll have, in each cell, not 14 possible crops but

$$\frac{n!}{k! \cdot (n-k)!} = \frac{14!}{4! \cdot (14-4)!} = 1001$$

So we'll have to consider for each cell 1001 different combinations of crops. Of course, this number will be smaller since not all the crops can grow in most of the cells, and that will reflect a real possible situation in the world.

9. Bibliography

Richard G. Allen, Luis S. Pereira, Dirk Raes, Martin Smith. FAO Irrigation and Drainage Paper No.56 - Crop Evapotranspiration. Italy: FAO, Water Resources, Development and Management Service, 2006.

Wolfram Mauser, Gernot Klepper, Florian Zabel, Ruth Delzeit, Tobias Hank, Birgitta Putzenlechner & Alvaro Calzadilla. Global biomass production potentials exceed expected future demand without the need for cropland expansion. http://www.nature.com/ncomms/2015/151112/ncomms9946/full/ncomms9946.html , 24th July 2015.

GAEZ – Global Agro-Ecological Zones. http://gaez.fao.org/ , Italy: FAO, 2015.

FAOSTAT - food and agriculture organization of the united nations statistics division. http://faostat3.fao.org/home/E , Italy: FAO, 2015.

M. Falkenmkar, J. Rockstörm. The New Blue and Green Water Paradigm: Breaking New Ground for Water Resources Planning and Management. US: Journal Of Water Resources Planning And Management – American Society of Civil Engineer's, June 2006.

M.M. Mekonnen, A. Y. Hoekstra. The green, blue and grey water footprint of crops and derived crop products. Hydrology and Earth System Sciences, 2011.

M. Tunietti, S. Tamea, P. D'Odorico, F. Laio, L. Ridolfi. Global sensitivity of high-resolution estimates of crop water footprint. Italy: Department of Environmental, Land, and Infrastructure Engineering, Politecnico di Torino; Department of Environmental Sciences, University of Virginia, 2015.

Nathaniel D. Mueller, James S. Gerber, Matt Johnston, Deepak K. Ray, Navin Ramankutty & Jonathan A. Foley. Closing yield gaps through nutrient and water management. US: Macmillan Publishers Limited, 2013.

Deepak K. Ray and Jonathan A. Foley. Increasing global crop harvest frequency: recent trends and future directions. US: Institute on the Environment, University of Minnesota, 325 Learning and Environmental Sciences, 6th June 2013.

David Tilman & Michael Clark. Global diets link environmental sustainability and human health.US: Macmillan Publishers Limited, 27th November 2014.

Samir Suweis, Joel A. Carr, Amos Maritan, Andrea Rinaldo and Paolo D'Odorico. Resilience and reactivity of global food security. http://www.pnas.org/content/112/22/6902.full , 15th April 2015.

Bender, R.R., J.W. Haegele, M.L. Ruffo, and F.E. Below. Nutrient uptake, partitioning, and remobilization in modern, transgenic insect-protected maize hybrids. US: American Society of Agronomy, 2013.

Ritchie, S.W., J.J. Hanway, and H.E. Thompson. How a Soybean Plant Develops. US: Iowa State University Coop. Ext. Special Report no. 53, 1982.

Salvagiotti, F., J.E. Specht, K.G. Cassman, D.T. Walters, A.Weiss, and A. Dobermann. Growth and nitrogen fixation in high-yielding soybean: Impact of nitrogen fertilization. US: American Society of Agronomy, 2009.

Salvagiotti, F., K.G. Cassman, J.E. Specht, D.T. Walters, A. Weiss, and A. Doberman. Nitrogen uptake, fixation, and response to fertilizer N in soybeans: A review. US: Department of Agronomy and Horticulture, University of Nebraska-Lincoln, 2008.

Schmidt, J.P., M.A. Schmitt, G.W. Randall, J.A. Lamb, J.H. Orf, and H. Gollany. Swine manure application to nodulating and non-nodulating soybean. US: American Society of Agronomy, 2000.

Thies, J.E., P.W. Singleton, and B.B. Bohlool. Phenology, growth, and yield of field-grown soybean and bush bean as a function of varying modes of N nutrition. Great Britain: Elsevier Science Ltd, 1995.

Pellegrino Principe. HTML5, CSS3 Javscript. Italy: Apogeo Edizioni, 2012.

Kevin Yank, G. Panico, S. Marconi. Sviluppare applicazioni con PHP e MySQL. Apogeo Edizioni, 2012.

Scott Granneman, R. Castaldo. Linux. US: Pearson Education, 2007.

Valerio Noti. GIS open source per geologia e ambiente. Analisi e gestione di dati territoriali e ambientali con QGIS. Dario Flaccovio Editore, 2014.

Joel Lawhead. QGIS Python Programming Cookbook: Over 140 Recipes to Help You Turn Qgis from a Desktop Gis Tool into a Powerful Automated Geospatial Framework. Packt open source, March 2015.

Appendix 1 – Crops List for WaterGrab

A full list of all the crops you can use for simulation inside WaterGrab.Zone project

CROP ID	CROP NAME	CROP ID	CROP NAME
809	Abaca (Manila Hemp)	333	Linseed
800	Agave Fibers nes	210	Lupins
641	Alfalfa for Forage+Silag	56	Maize
221	Almonds	636	Maize for Forage+Silage
711	Anise, Badian, Fennel	571	Mangoes
515	Apples	299	Melonseed
526	Apricots	79	Millet
226	Areca Nuts (Betel)	103	Mixed Grain
366	Artichokes	292	Mustard Seed
367	Asparagus	839	Natural Gums
572	Avocados	836	Natural Rubber
203	Bambara Beans	702	Nutmeg, Mace, Cardamons
486	Bananas	234	Nuts nes
44	Barley	75	Oats
176	Beans, Dry	254	Oil Palm Fruit
414	Beans, Green	339	Oilseeds nes
647	Beets for Fodder	430	Okra
558	Berries nes	260	Olives
552	Blueberries	403	Onions, Dry
216	Brazil Nuts	402	Onions+Shallots, Green
181	Broad Beans, Dry	490	Oranges
420	Broad Beans, Green	600	Papayas
89	Buckwheat	534	Peaches and Nectarines
644	Cabbage for Fodder	521	Pears
358	Cabbages	187	Peas, Dry
101	Canary Seed	417	Peas, Green
568	Cantaloupes&oth Melons	687	Pepper,White/Long/Black

461	Carobs	748	Peppermint
426	Carrots	197	Pigeon Peas
648	Carrots for Fodder	689	Pimento, Allspice
217	Cashew Nuts	574	Pineapples
591	Cashewapple	223	Pistachios
125	Cassava	489	Plantains
265	Castor Beans	536	Plums
393	Cauliflower	296	Poppy Seed
108	Cereals nes	116	Potatoes
531	Cherries	211	Pulses nes
220	Chestnuts	394	Pumpkins, Squash, Gourds
191	Chick-Peas	92	Quinoa
401	Chillies&Peppers, Green	788	Ramie
693	Cinnamon (Canella)	270	Rapeseed
512	Citrus Fruit nes	547	Raspberries
640	Clover for Forage+Silage	27	Rice, Paddy
698	Cloves, Whole+Stems	149	Roots and Tubers nes
661	Cocoa Beans	71	Rye
249	Coconuts	638	Rye Grass,Forage+Silage
656	Coffee, Green	280	Safflower Seed
195	Cow Peas, Dry	328	Seed Cotton
554	Cranberries	289	Sesame Seed
397	Cucumbers and Gherkins	789	Sisal
550	Currants	83	Sorghum
577	Dates	637	Sorghum for Forage+Silag
399	Eggplants	530	Sour Cherries
821	Fibre Crops nes	236	Soybeans
569	Figs	723	Spices nes
773	Flax Fibre and Tow	373	Spinach
94	Fonio	541	Stone Fruit nes, Fresh
619	Fruit Fresh nes	544	Strawberries
603	Fruit Tropical Fresh nes	423	String Beans
406	Garlic	157	Sugar Beets
720	Ginger	156	Sugar Cane
549	Gooseberries	161	Sugar Crops nes
507	Grapefruit and Pomelos	267	Sunflower Seed
560	Grapes	649	Swedes for Fodder
639	Grasses nes,Forage+Silag	122	Sweet Potatoes
446	Green Corn (Maize)	495	Tang.Mand.Clement.Satsma
642	Green Oilseeds fr Fodder	136	Taro (Coco Yam)
242	Groundnuts in Shell	674	Tea
225	Hazelnuts (Filberts)	667	Tea nes
777	Hemp Fibre and Tow	826	Tobacco Leaves

336	Hempseed	388	Tomatoes
677	Hops	1	Trees
277	Jojoba Seeds	97	Triticale
780	Jute	646	Turnips for Fodder
782	Jute-Like Fibres	692	Vanilla
310	Kapok Fruit	463	Vegetables Fresh nes
592	Kiwi Fruit	655	Vegetables+Roots,Fodder
224	Kolanuts	205	Vetches
407	Leeks and Oth.Alliac.Veg	222	Walnuts
643	Leguminous nes,For+Sil	567	Watermelons
497	Lemons and Limes	15	Wheat
201	Lentils	137	Yams
372	Lettuce	135	Yautia (Cocoyam)

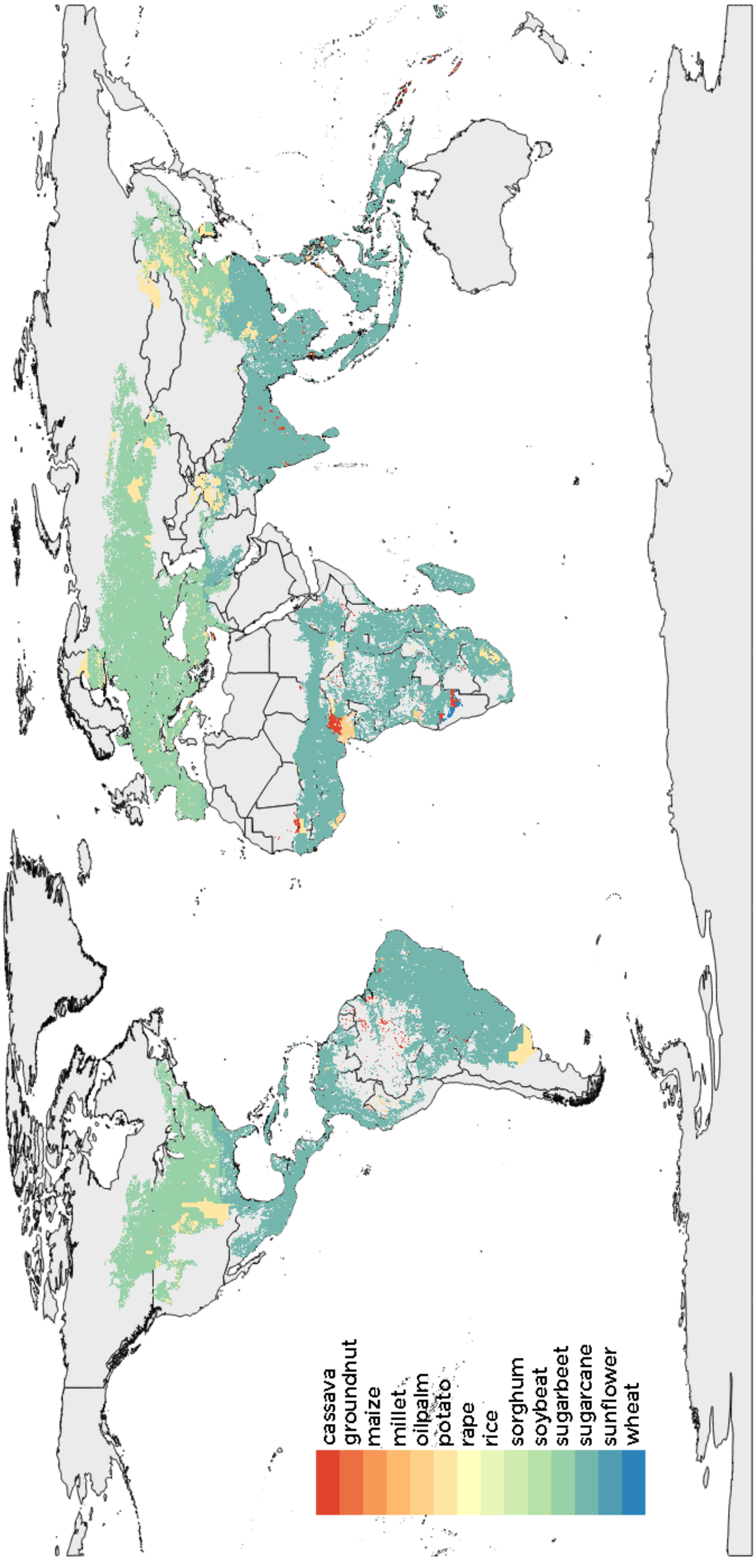
Appendix 2 – Maps from yield without borders

In the next pages you'll find all the maps resulted from our studies. For each criteria we'll have the crop distribution that resulted, as well as the values of $[tons]$, $[Gcal]$ and $[kg_{prot}]$ produced, as well as $[m^3_{water}]$ used.

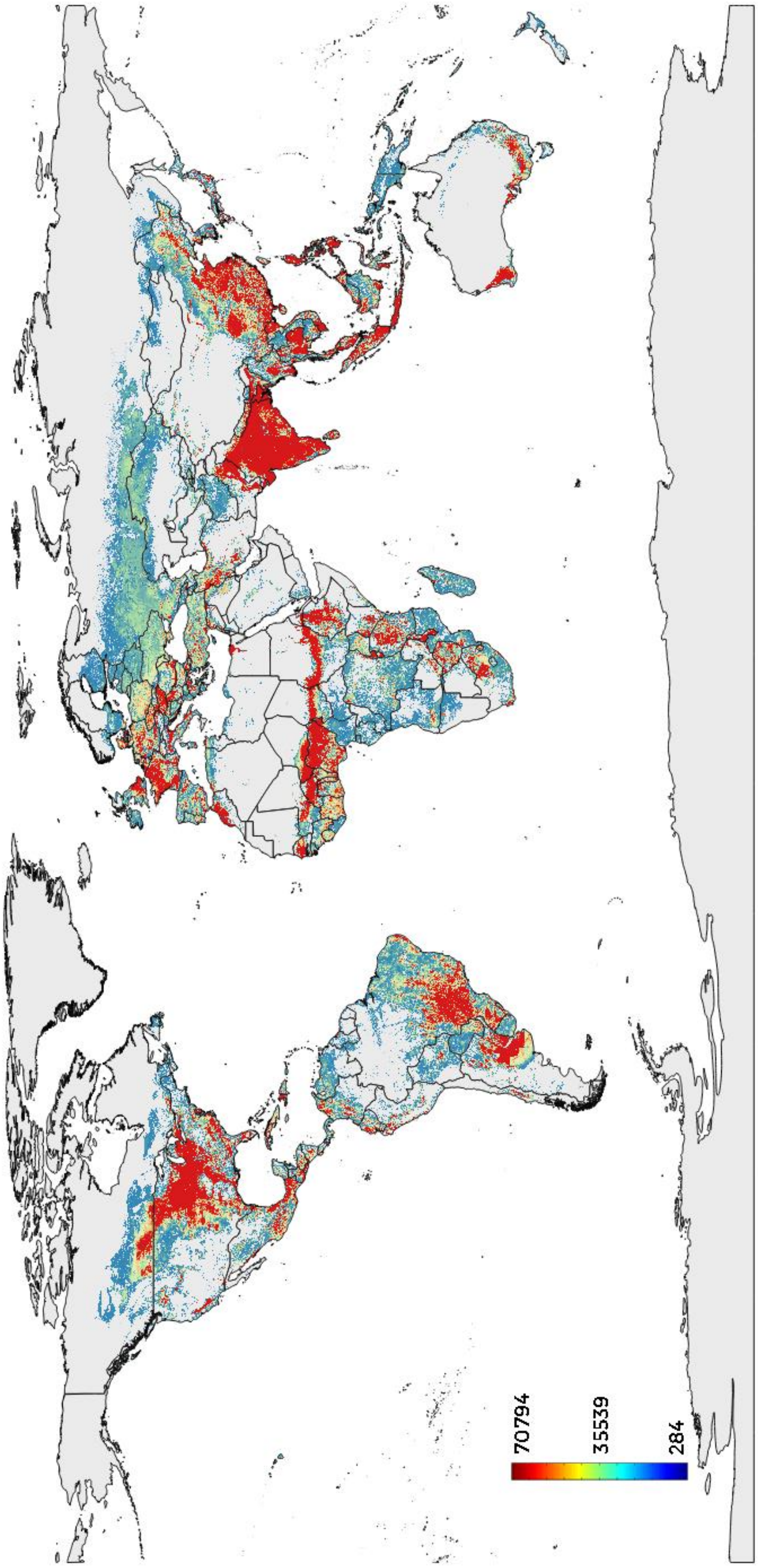
The results are considering maps derived from yield without taking under consideration the borders between countries. As we've seen, there's no much difference between the situation with and without borders.

Maximize kg production

Crop distribution

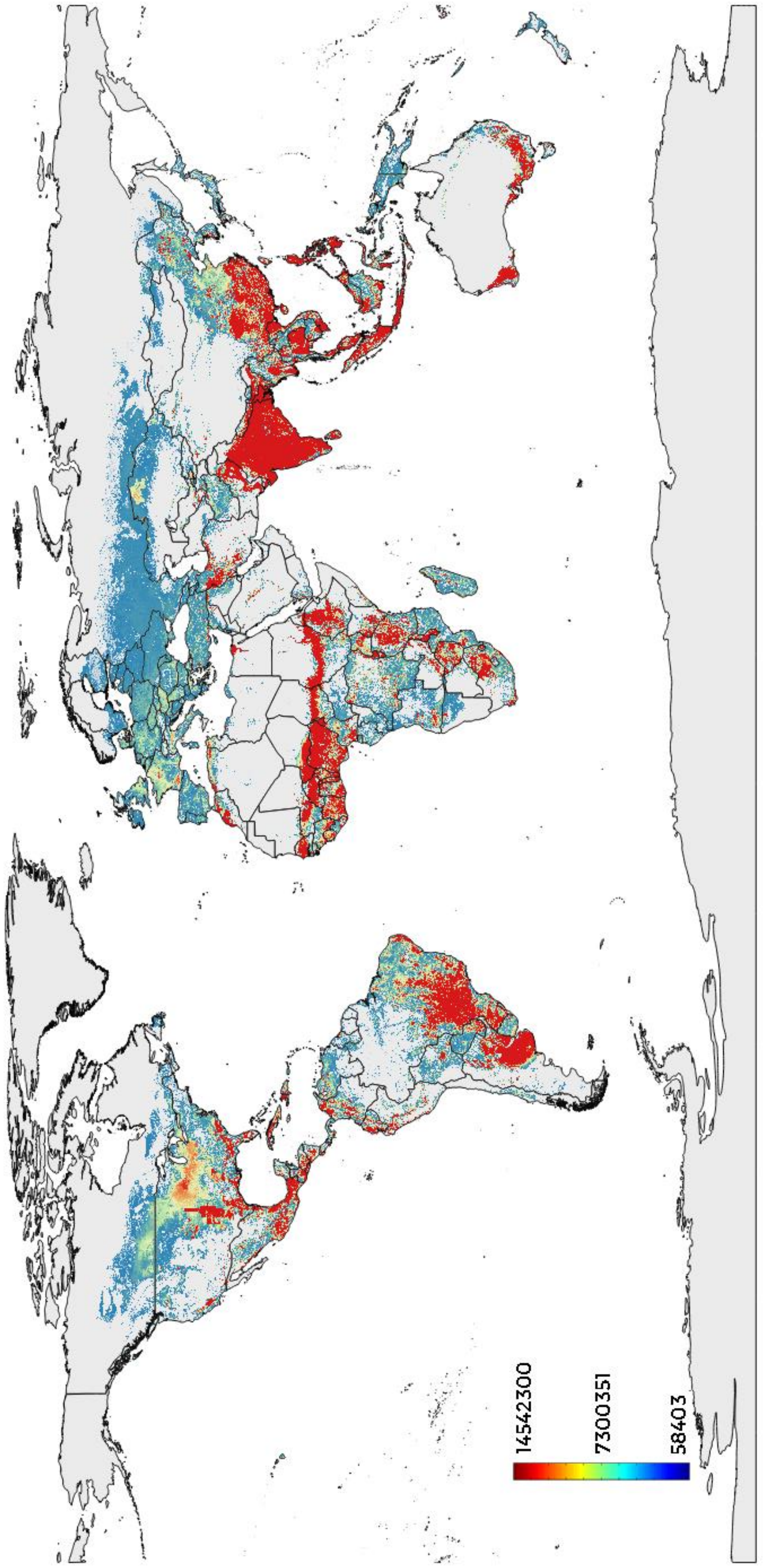


ton ***produced***

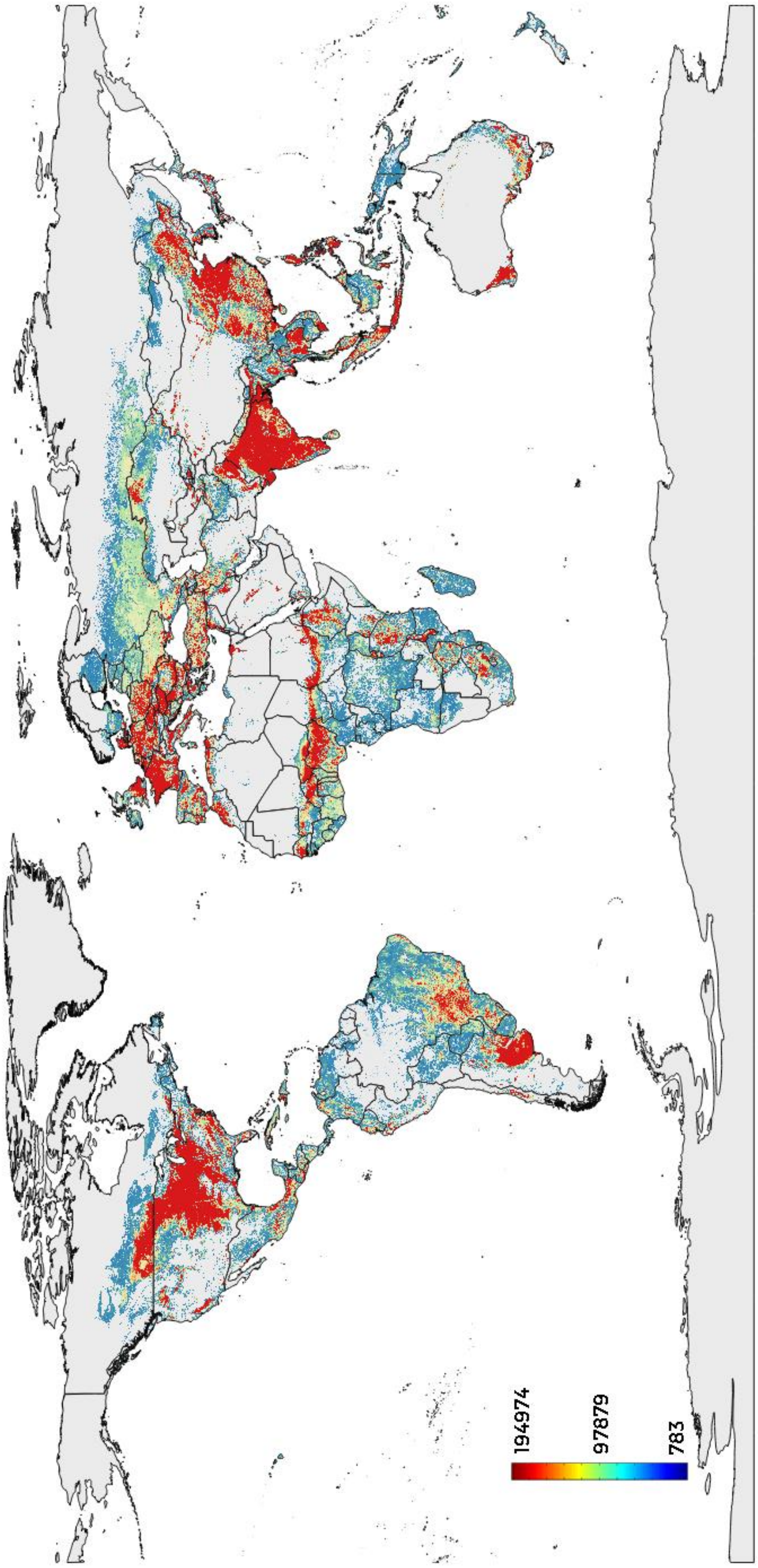


Maximize kg production

$Gcal$ *produced*

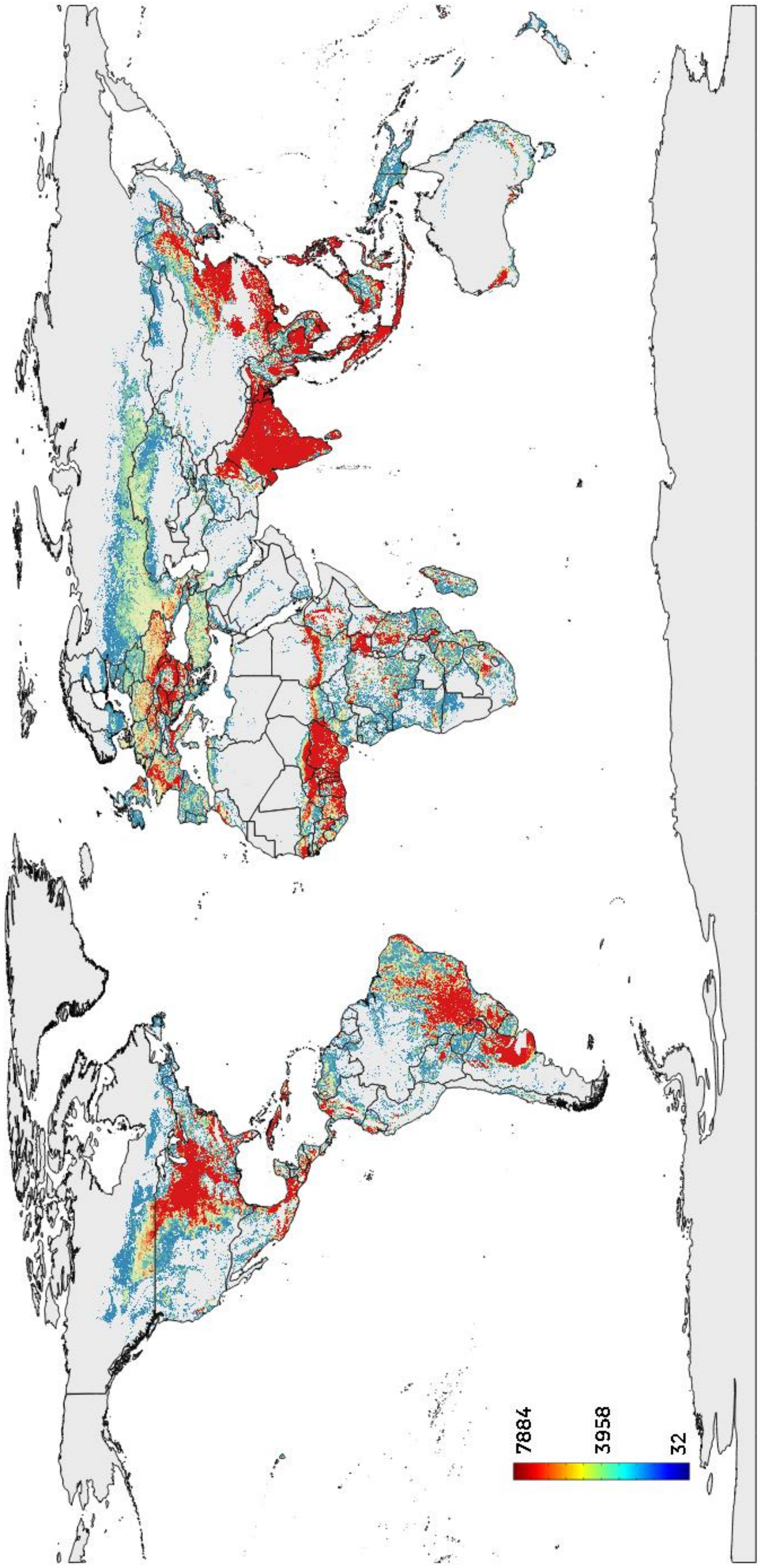


kg_{prot} ***produced***



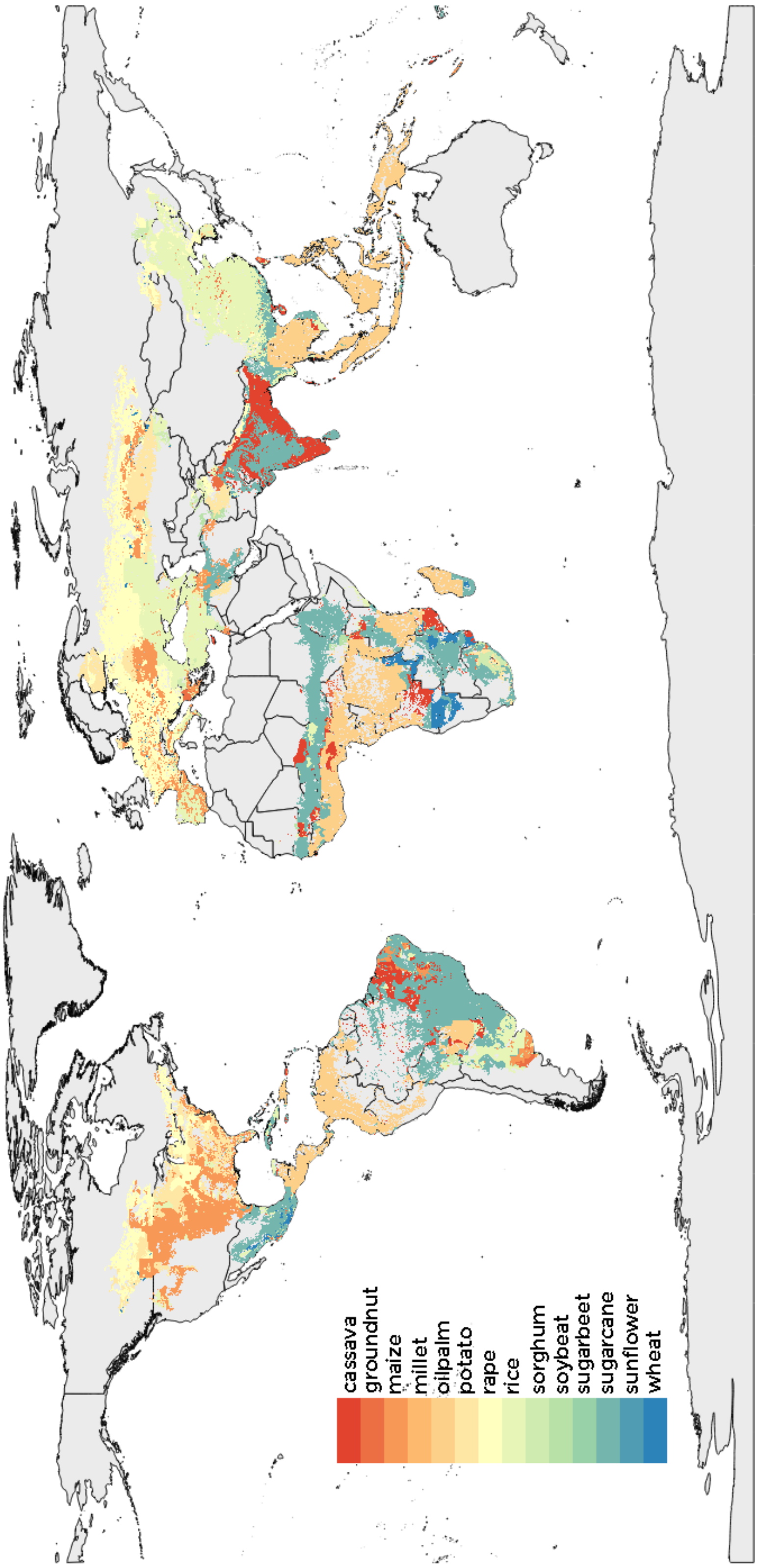
Maximize kg production

m^3_{water} ***used***

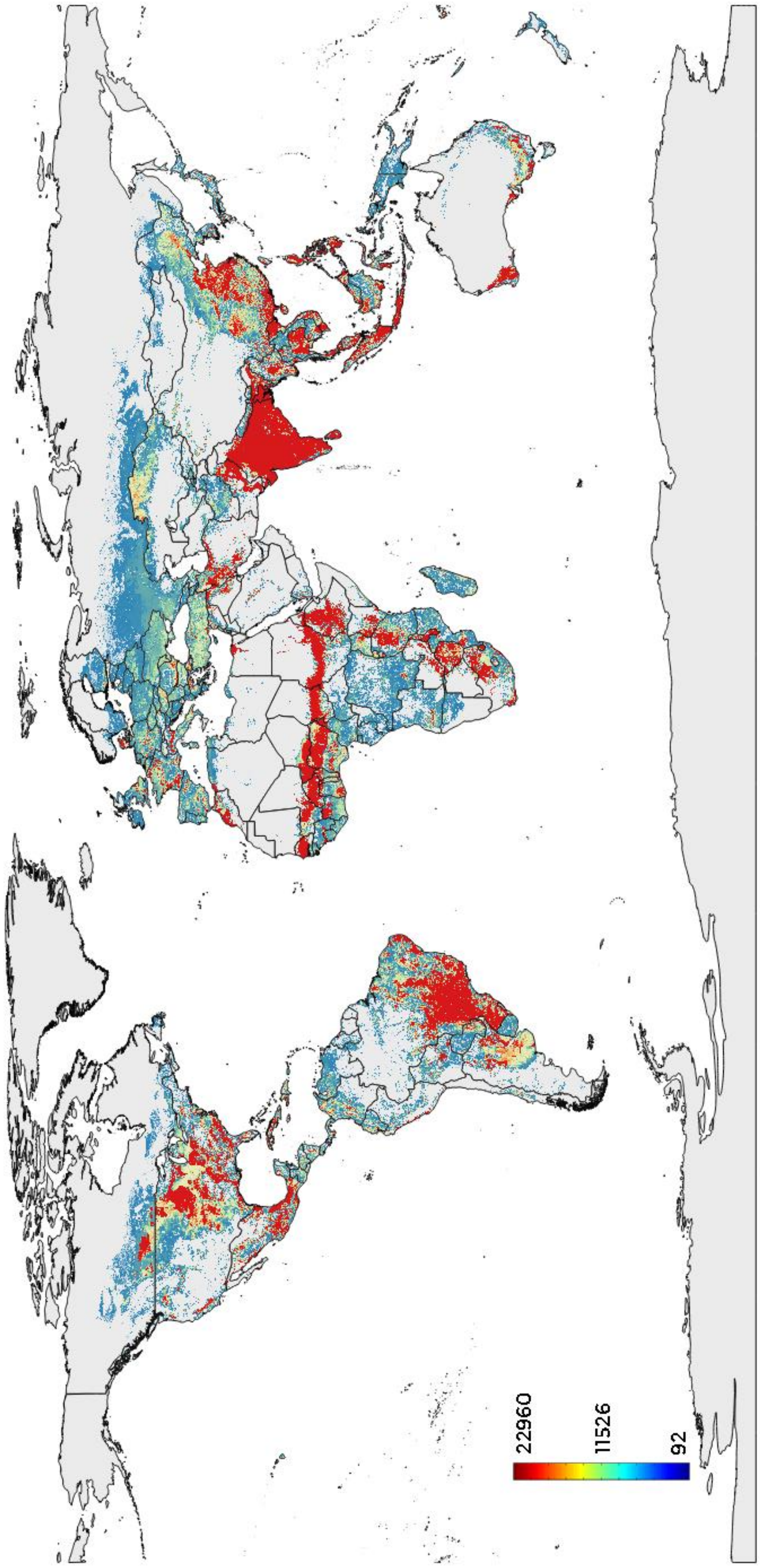


Maximize $kcal$ production

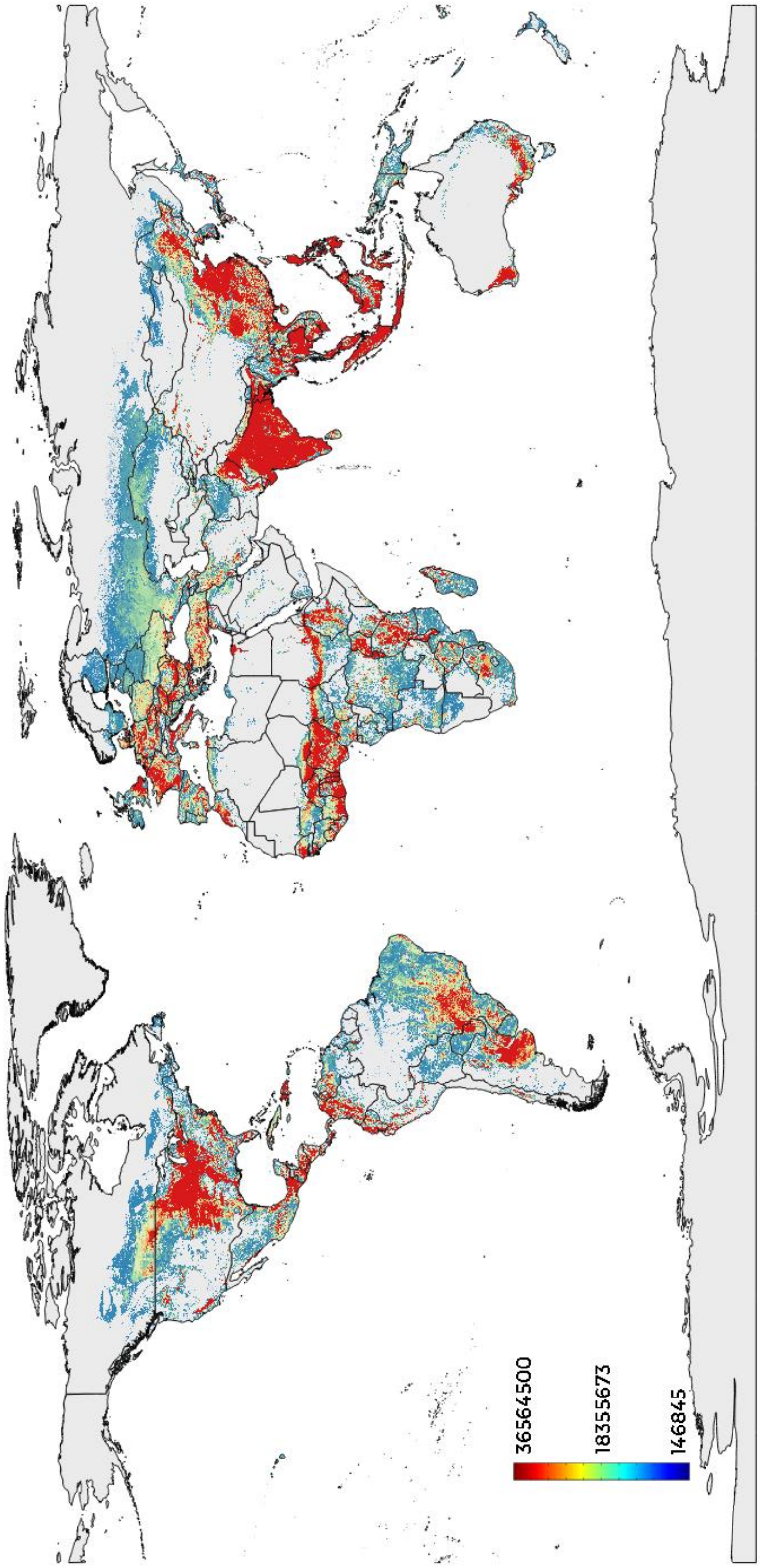
Crop distribution



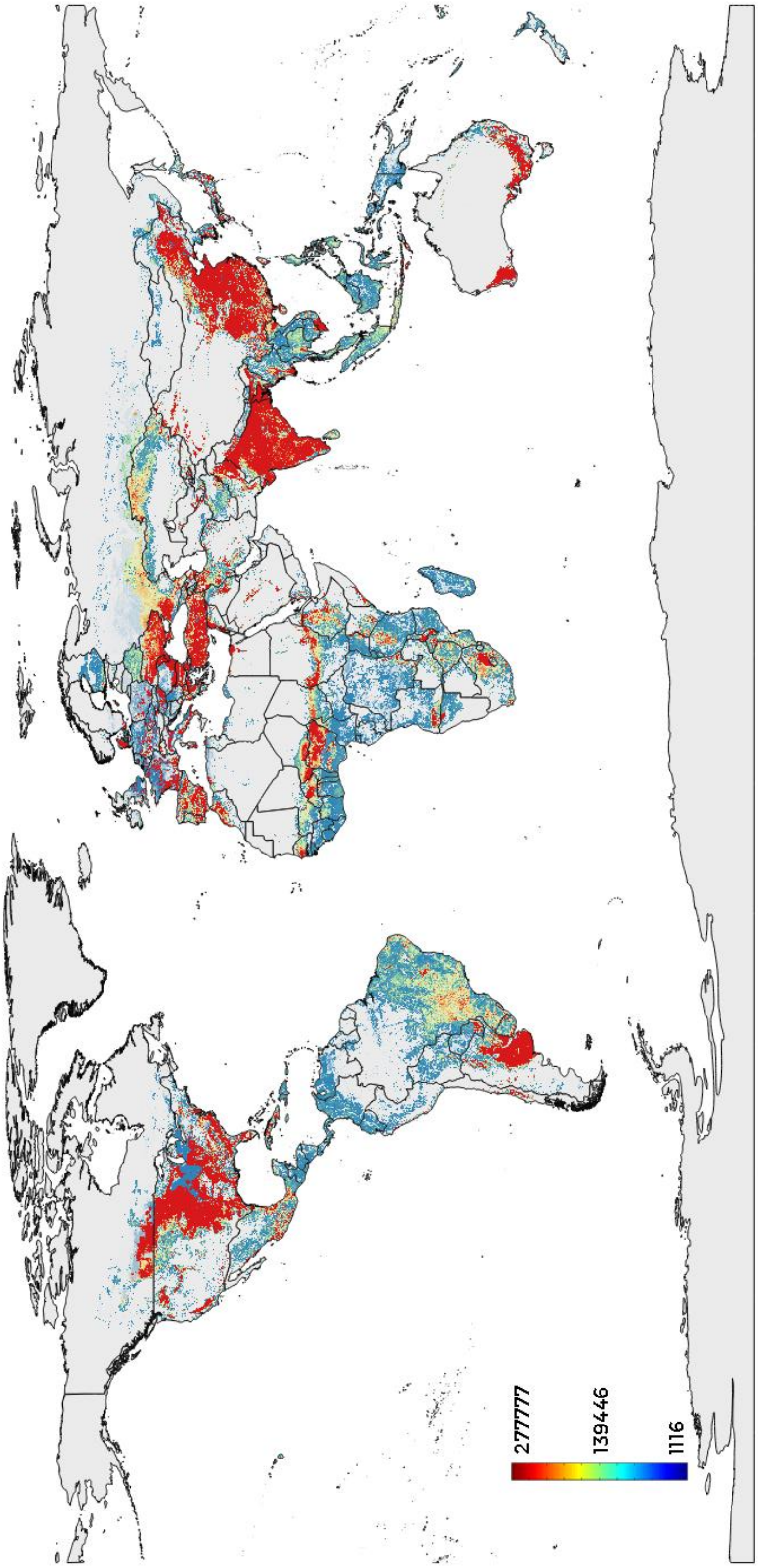
ton produced



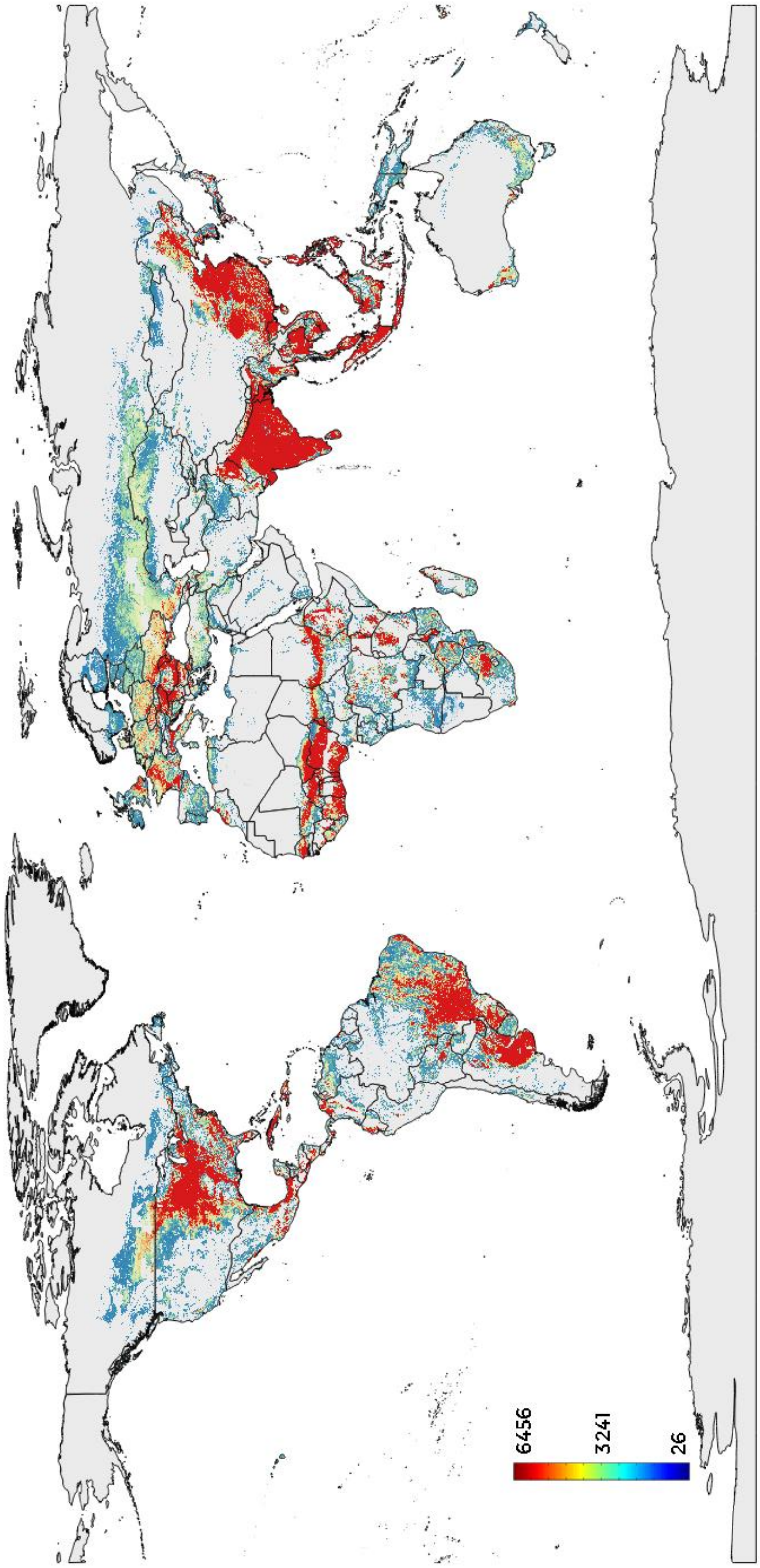
$Gcal$ ***produced***



kg_{prot} produced

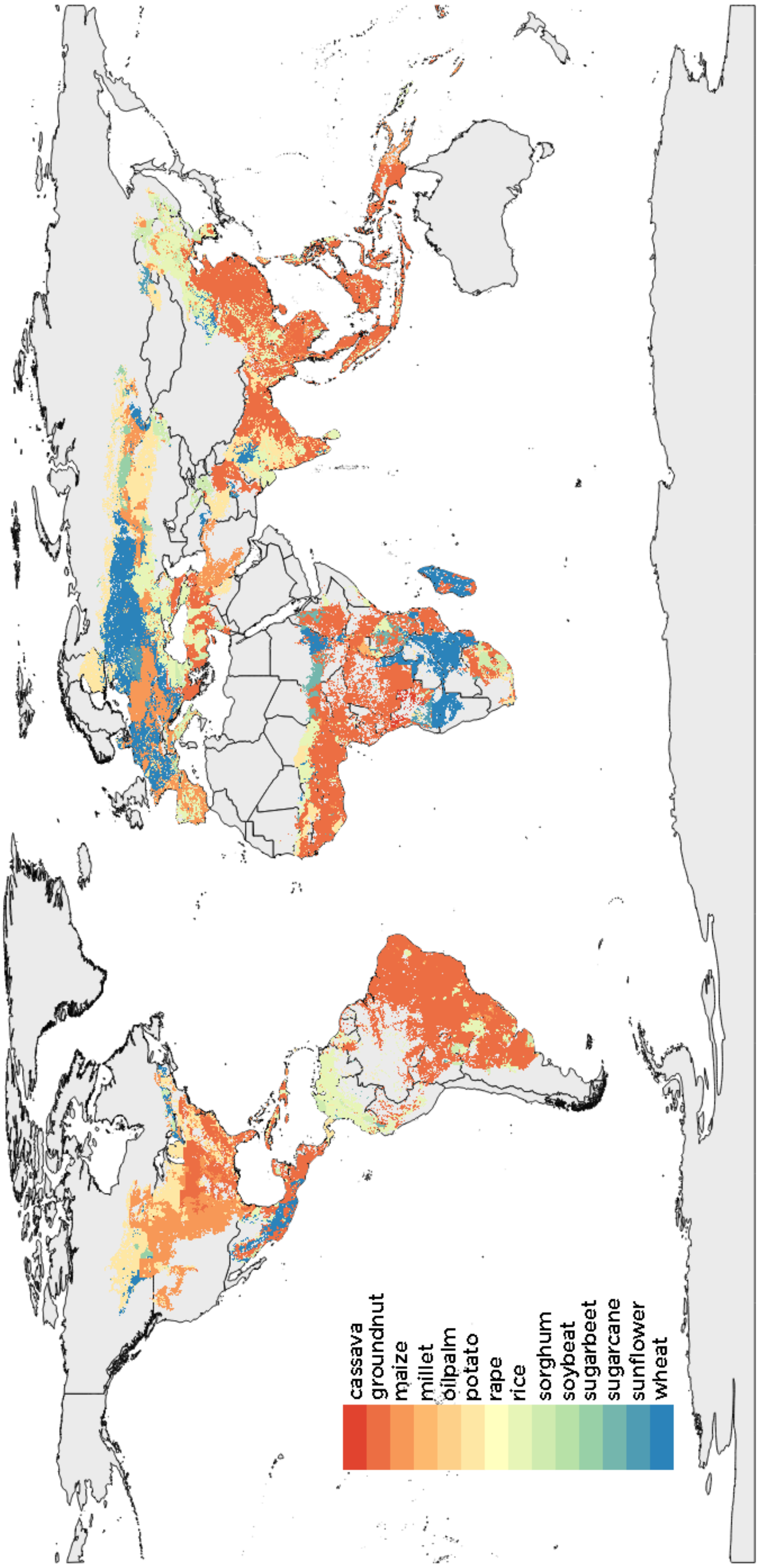


m^3_{water} ***used***



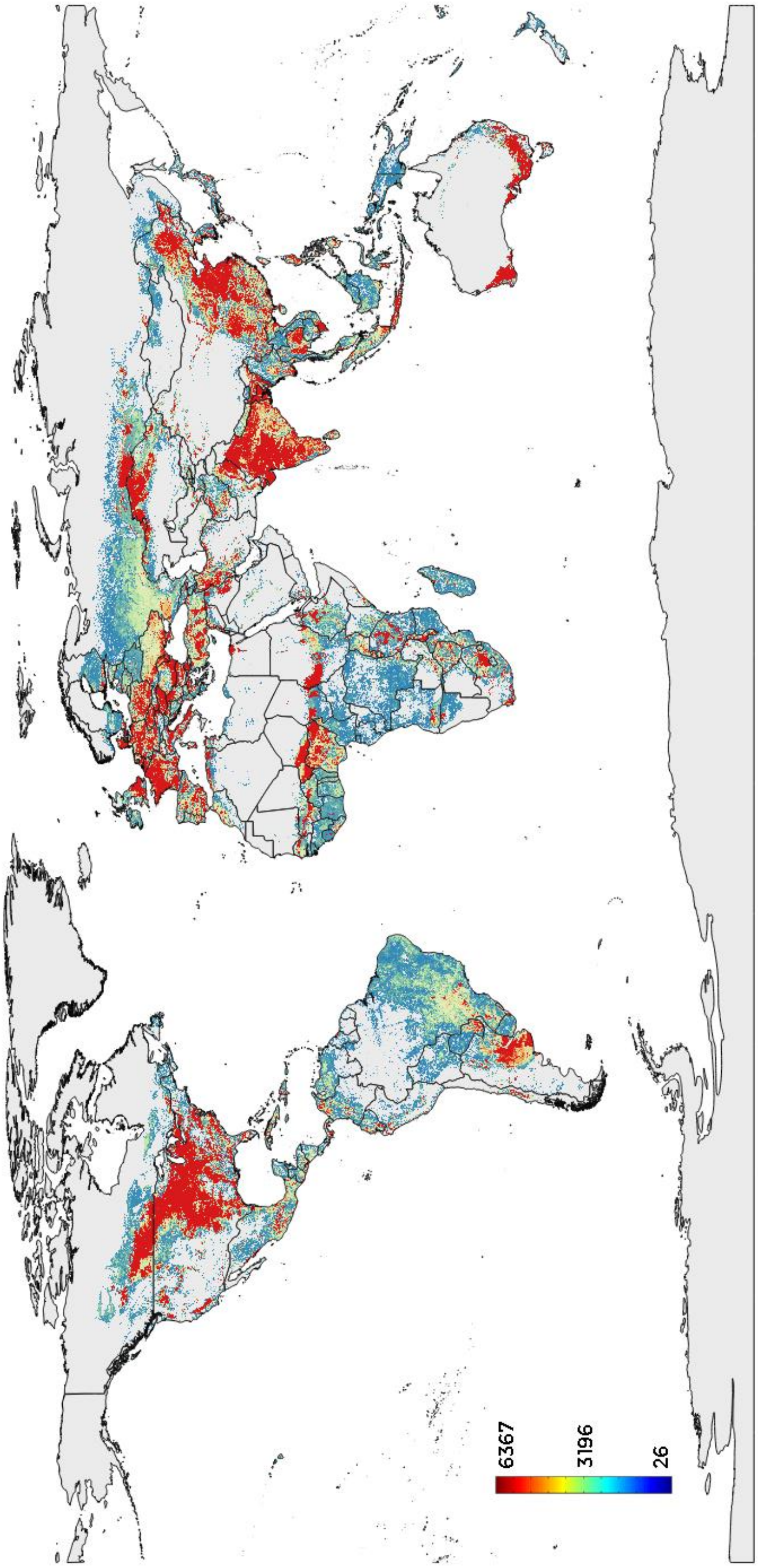
Maximize gr_{prot}production

Crop distribution

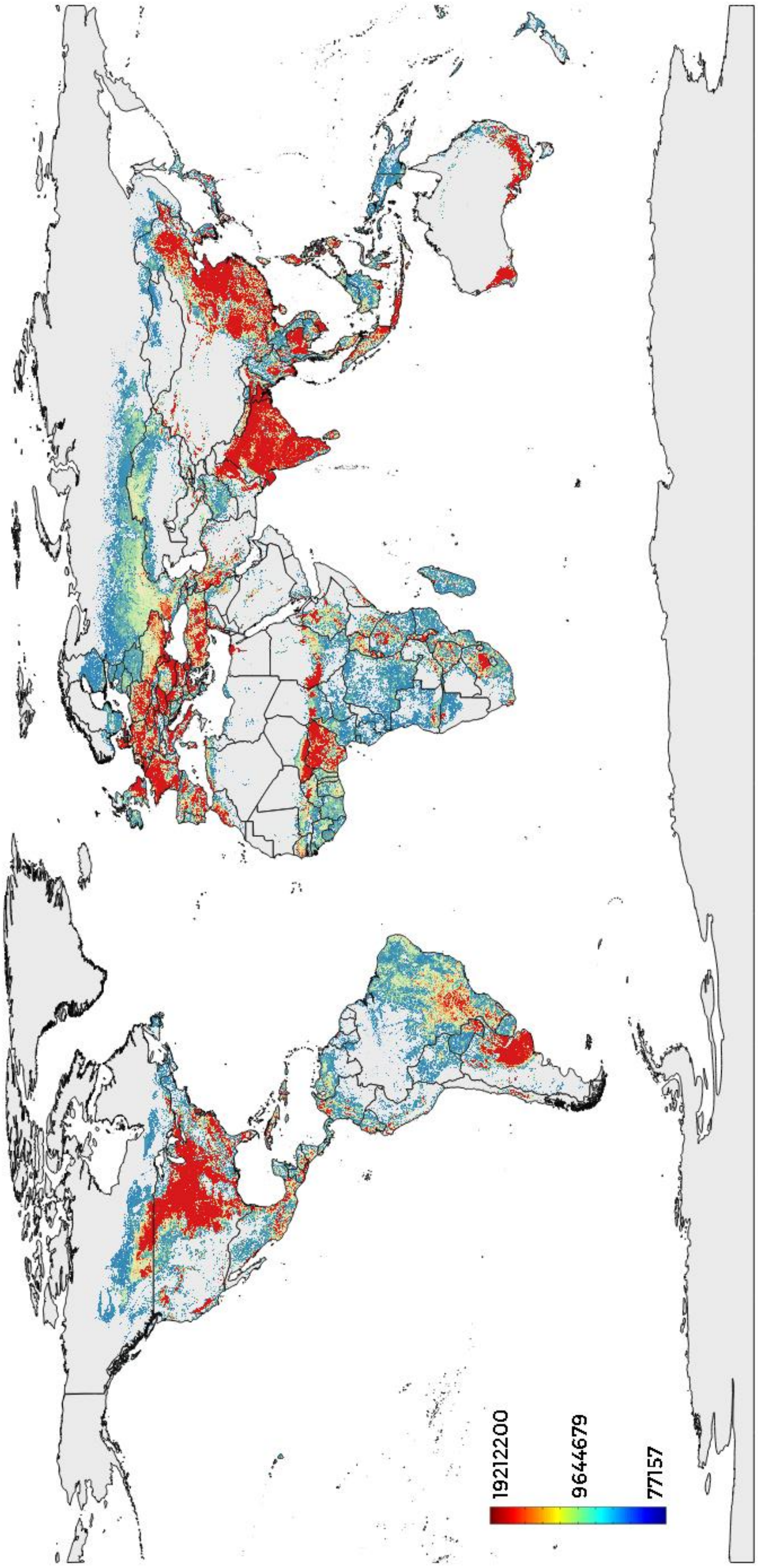


Maximize $grprot$production

ton produced

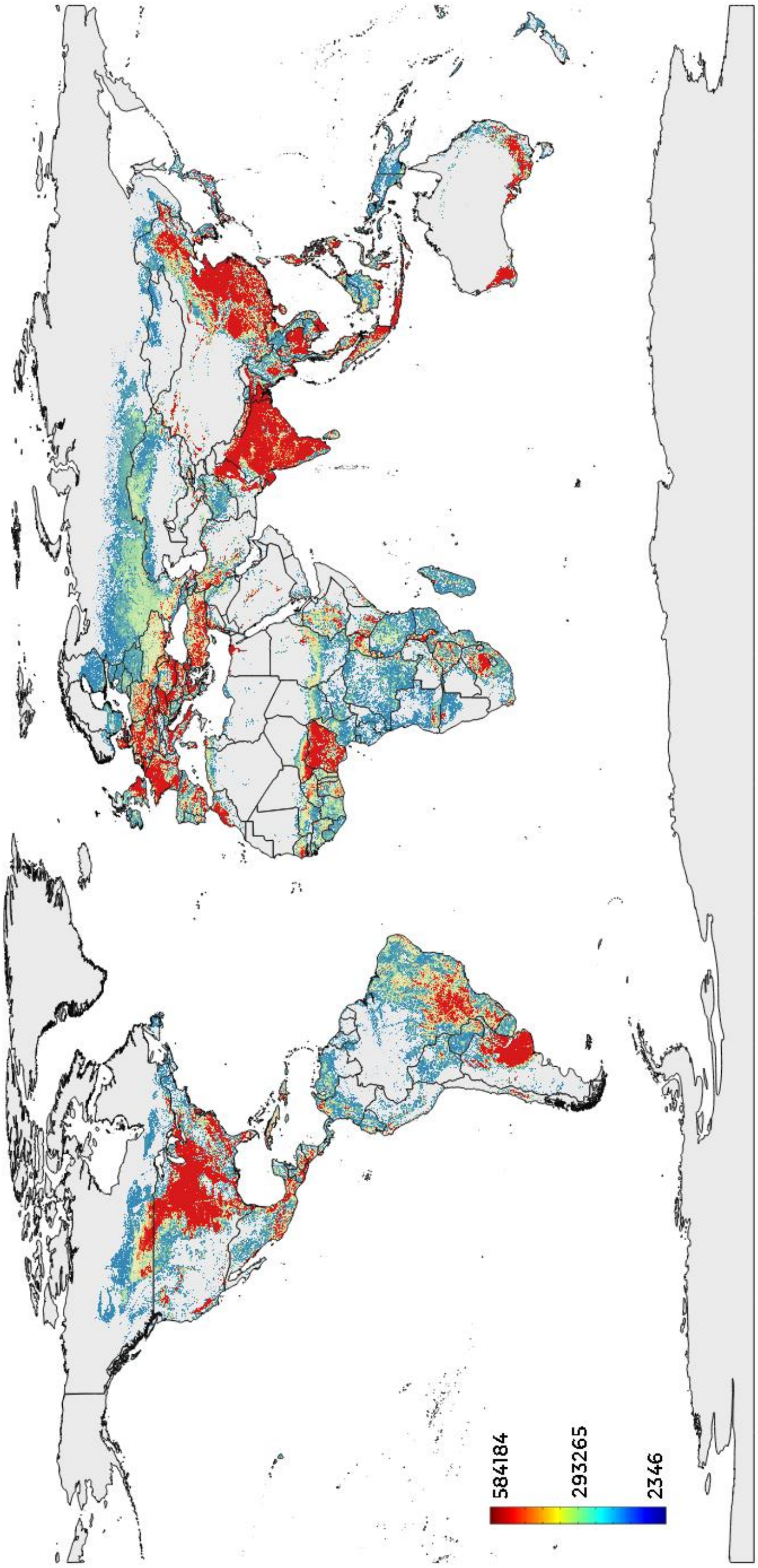


$Gcal$ ***produced***

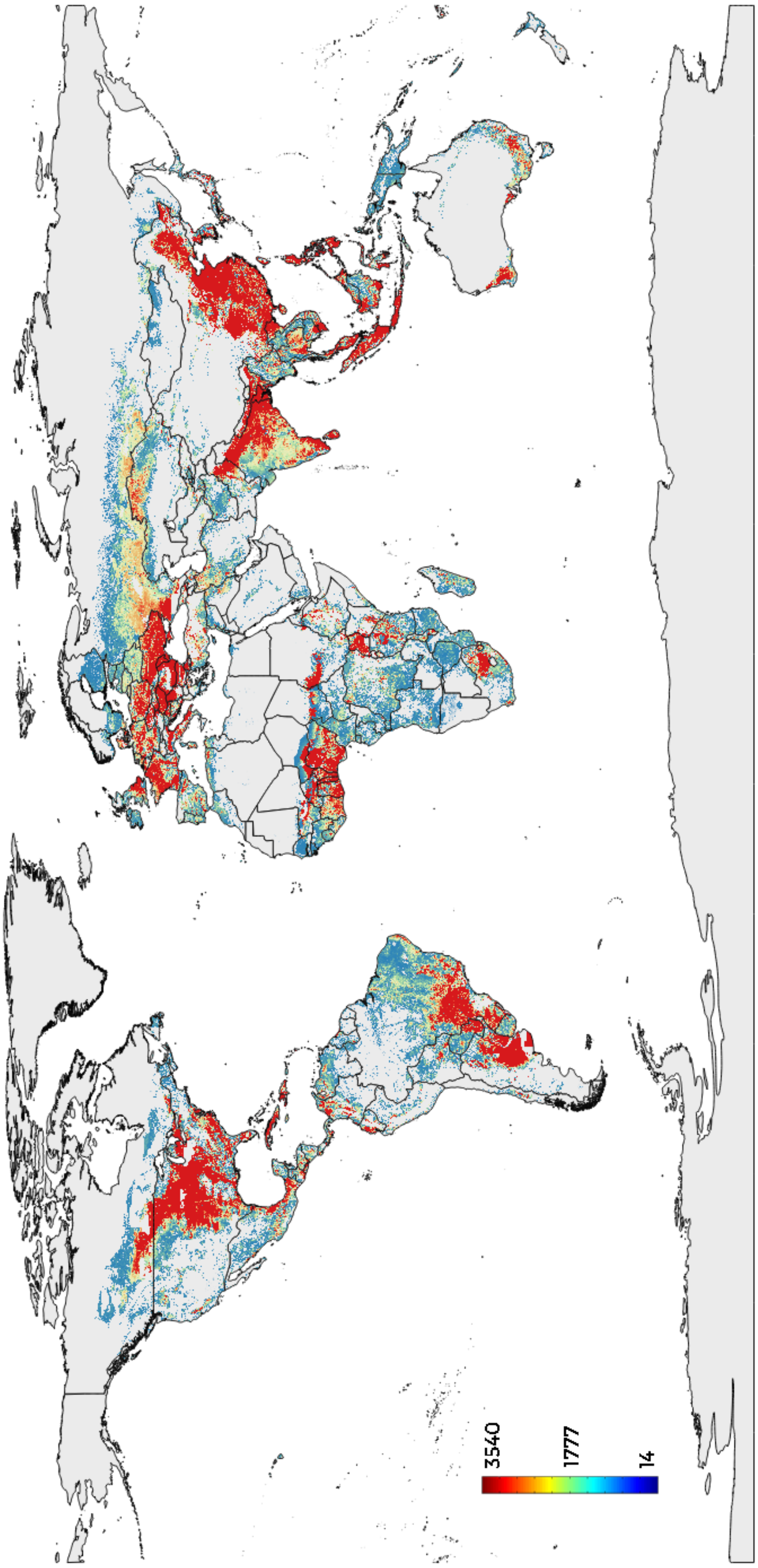


Maximize grprotproduction

kg_{prot} ***produced***

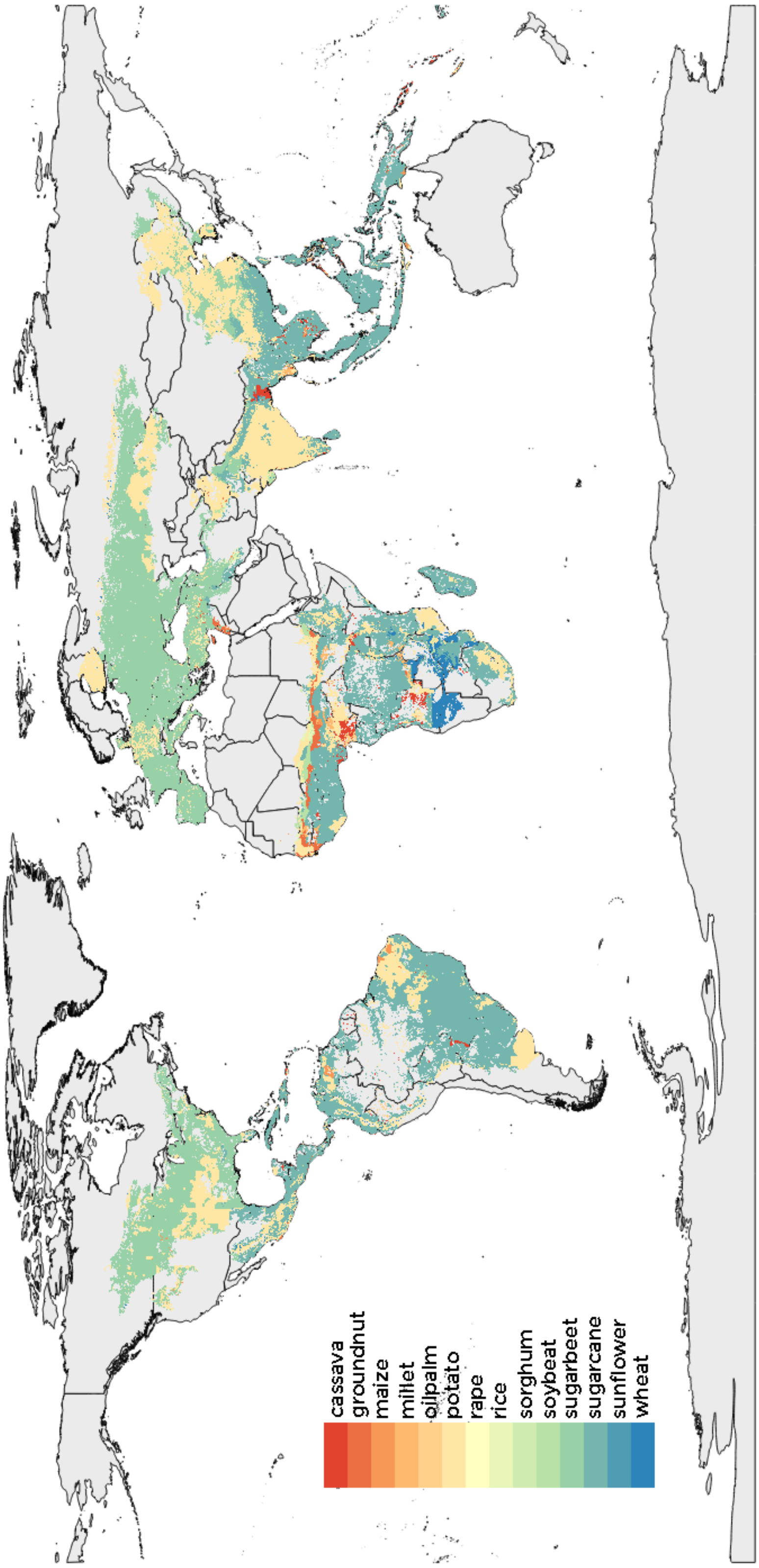


m^3_{water} ***used***

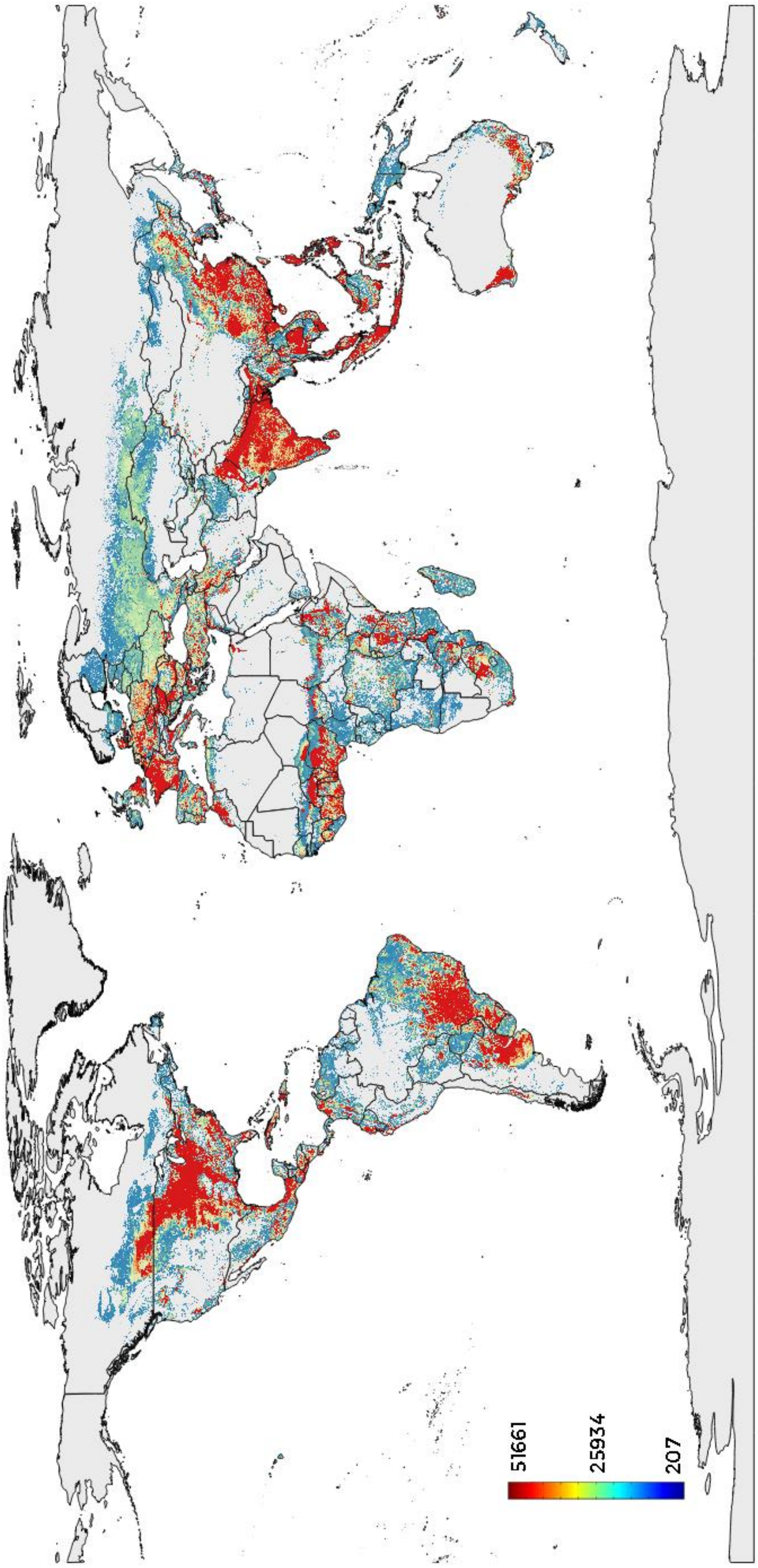


Minimize m_{water}^{3} utilized, maximizing kg production

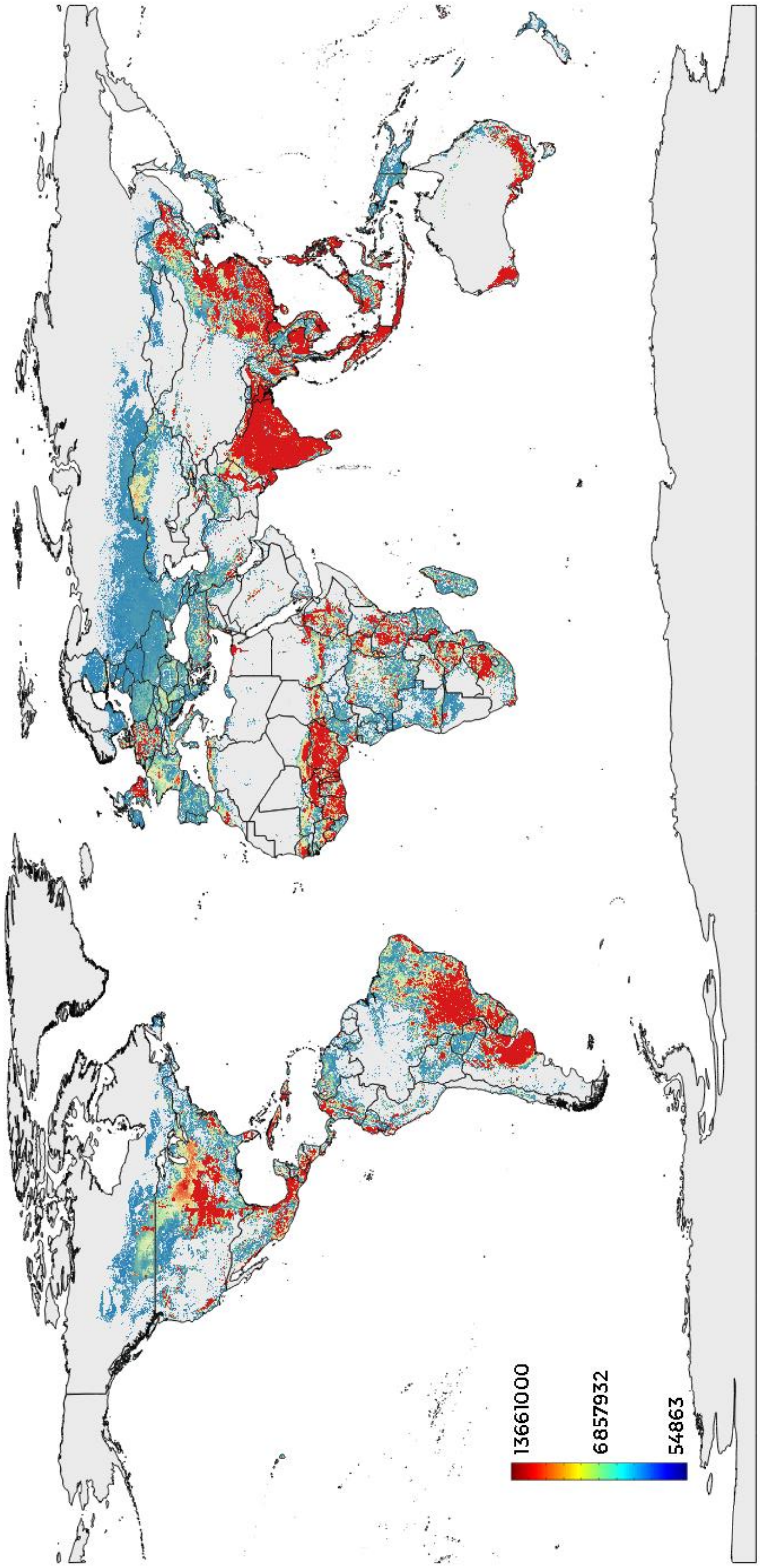
Crop distribution



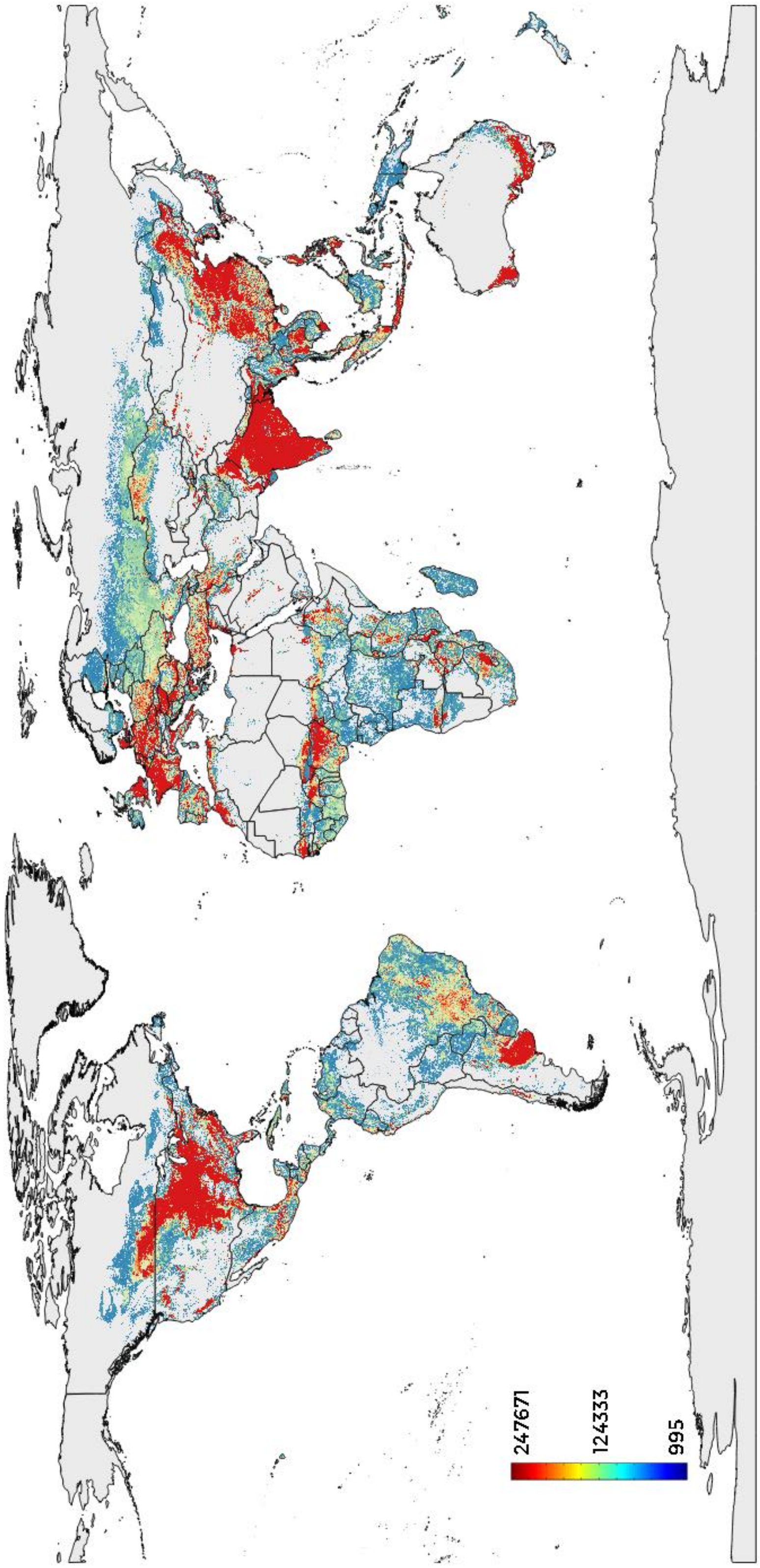
ton *produced*



$Gcal$ ***produced***

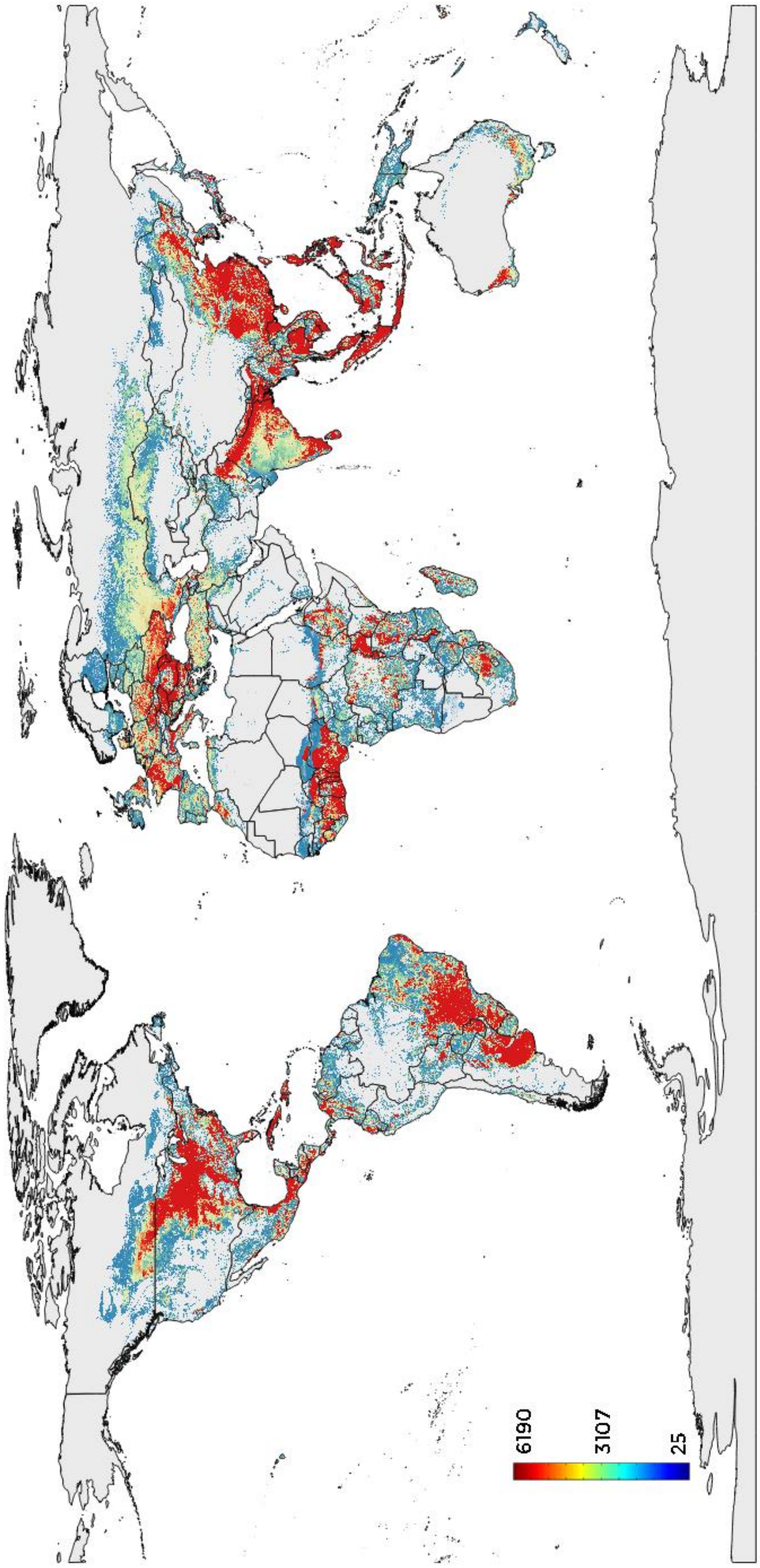


kg_{prot} ***produced***



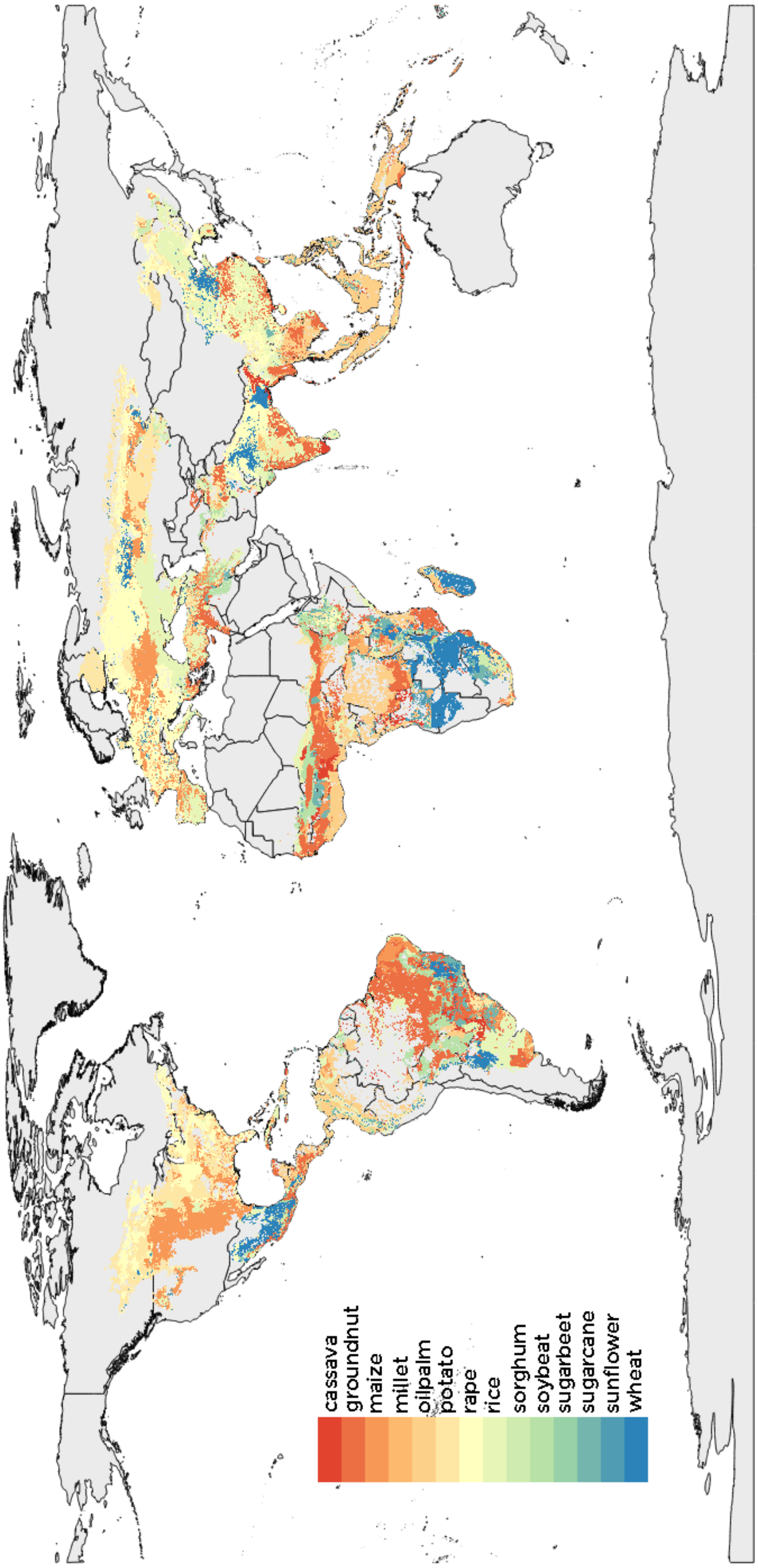
Minimize $mwater3$ utilized, maximizing kg production

m^3_{water} ***used***



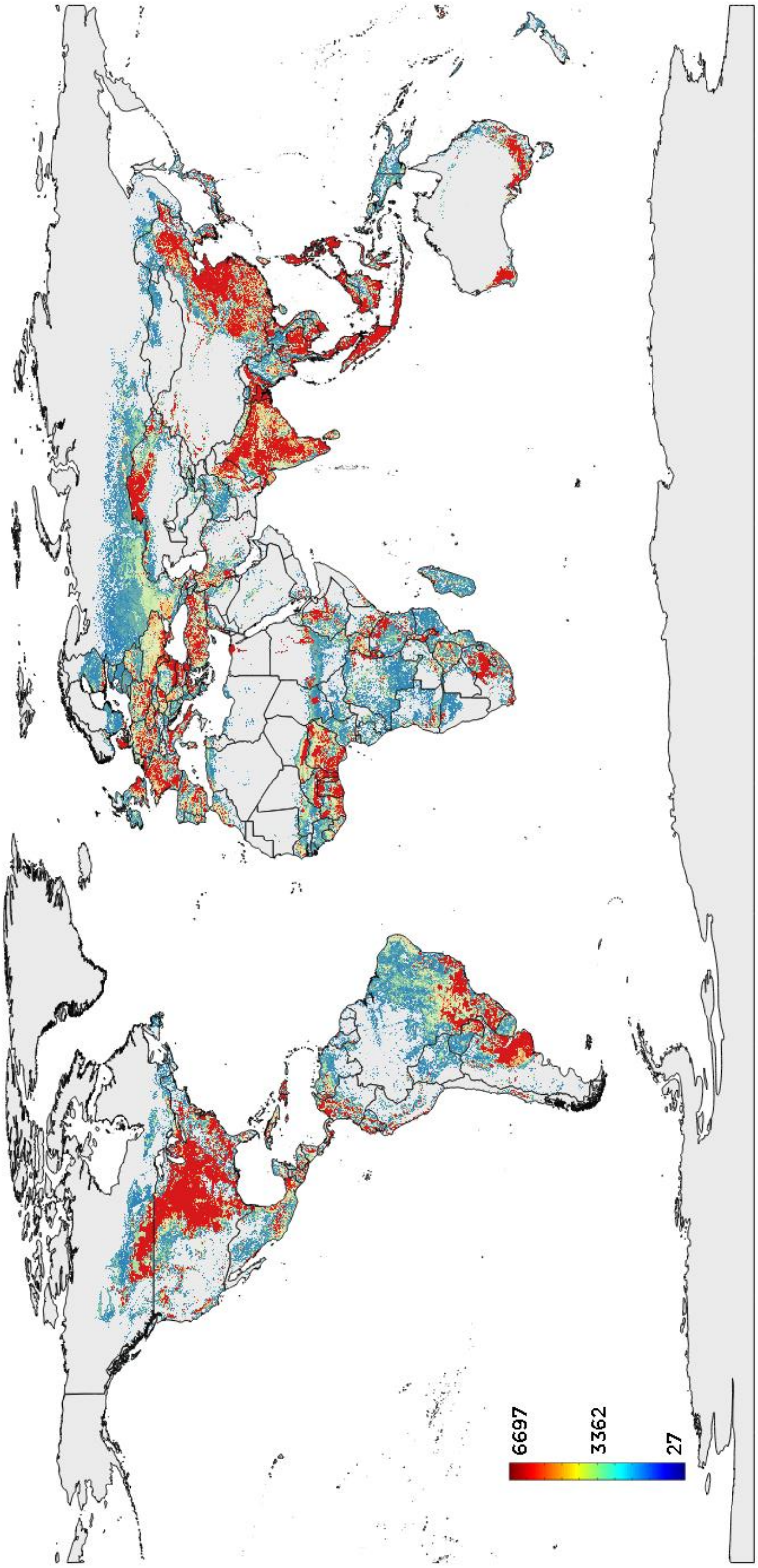
Minimize m^3_{water} utilized, maximizing $kcal$ production

Crop distribution



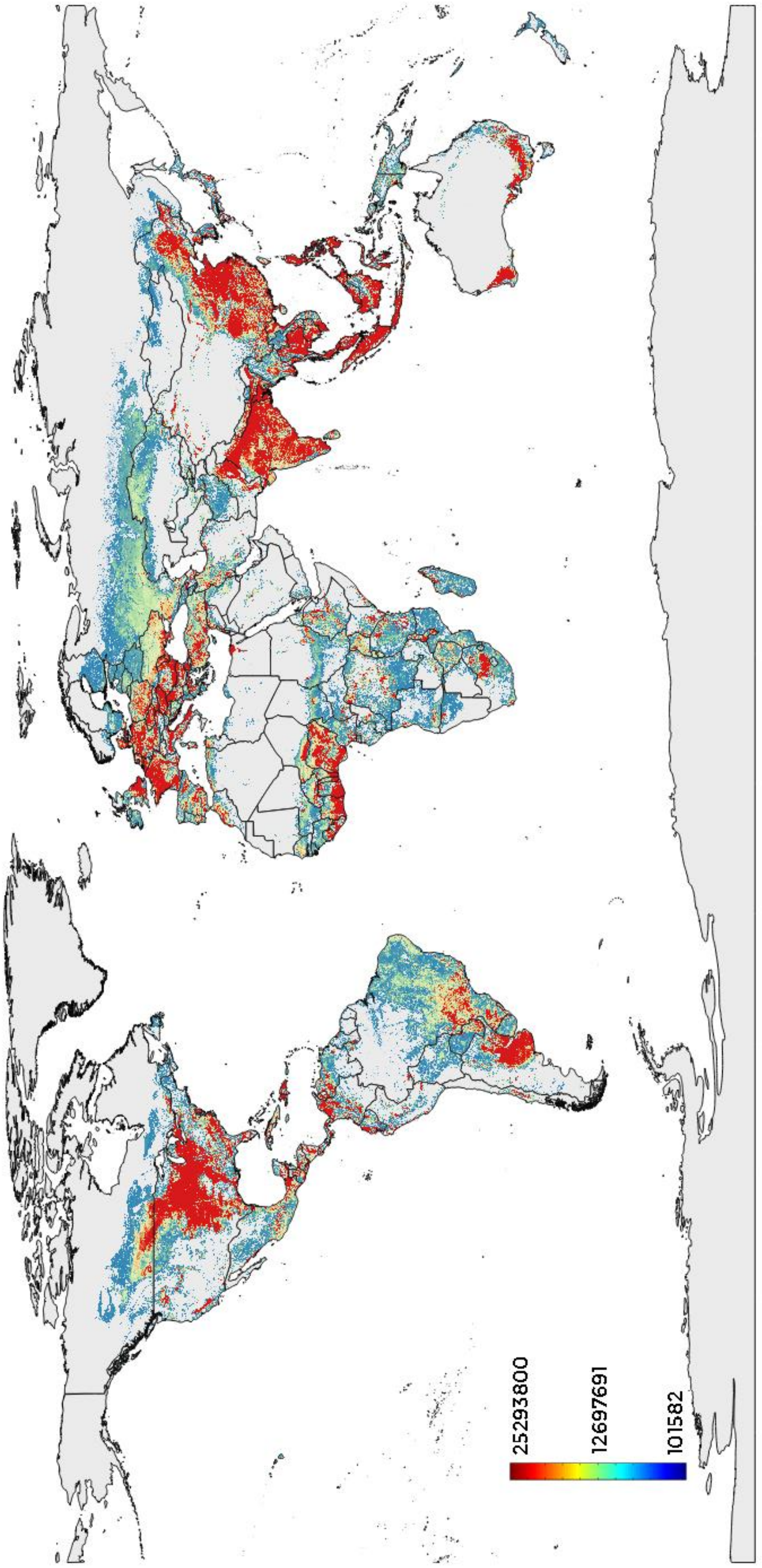
Minimize $mwater3$ utilized, maximizing $kcal$ production

ton ***produced***

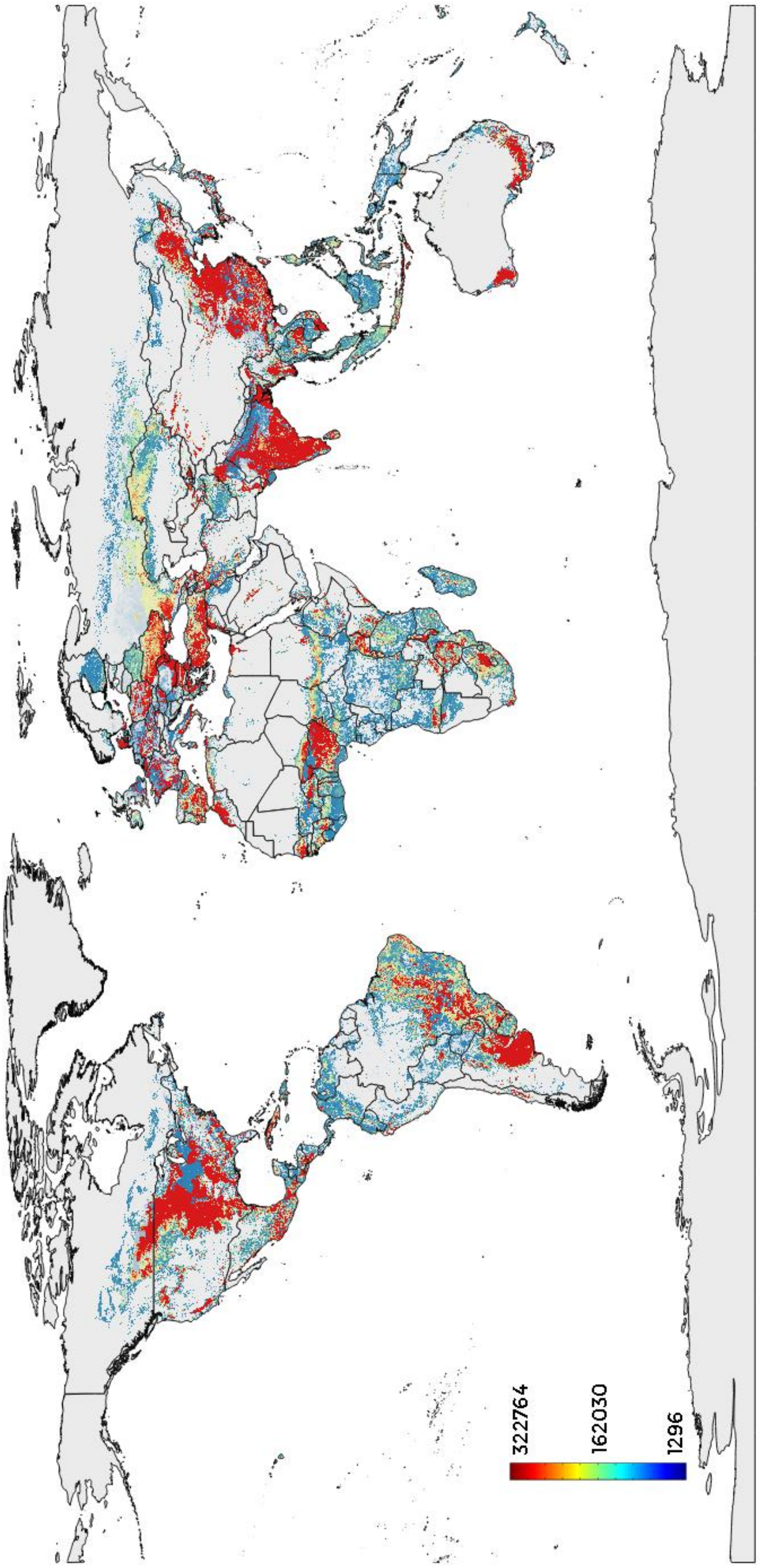


Minimize $mwater3$ utilized, maximizing $kcal$ production

$Gcal$ ***produced***

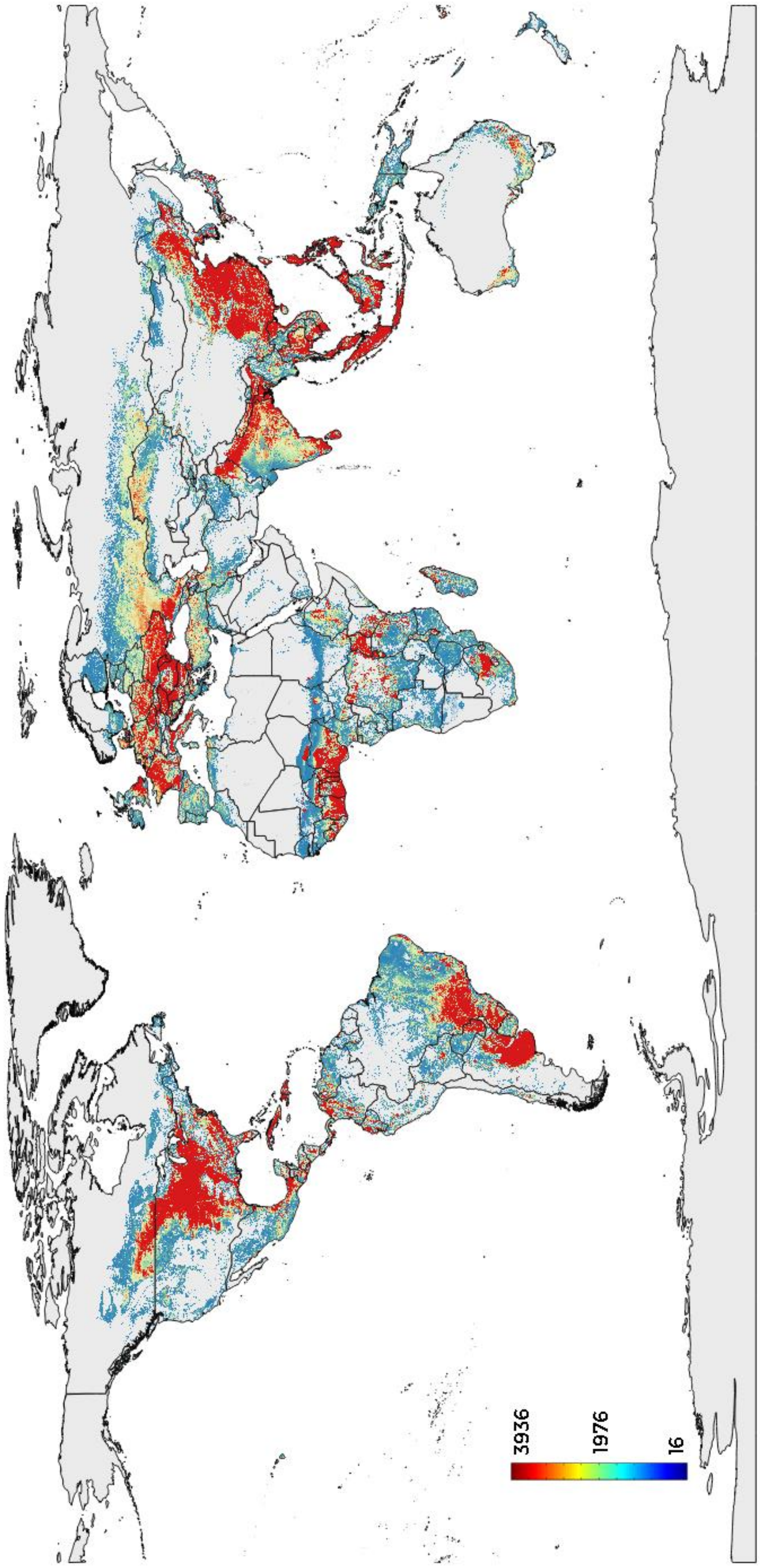


kg_{prot} ***produced***



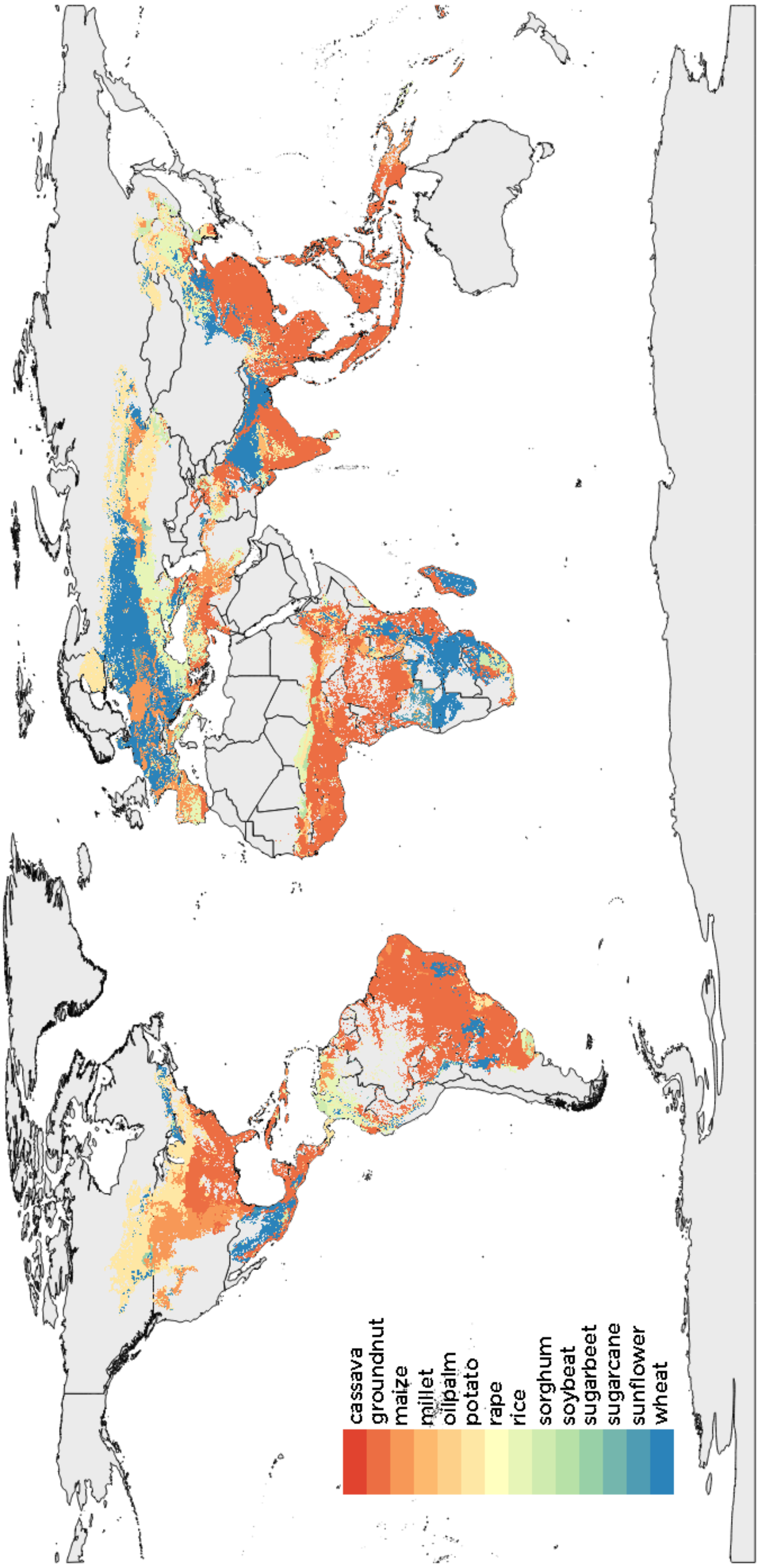
Minimize $mwater3$ utilized, maximizing $kcal$ production

m_{water}^{3} ***used***



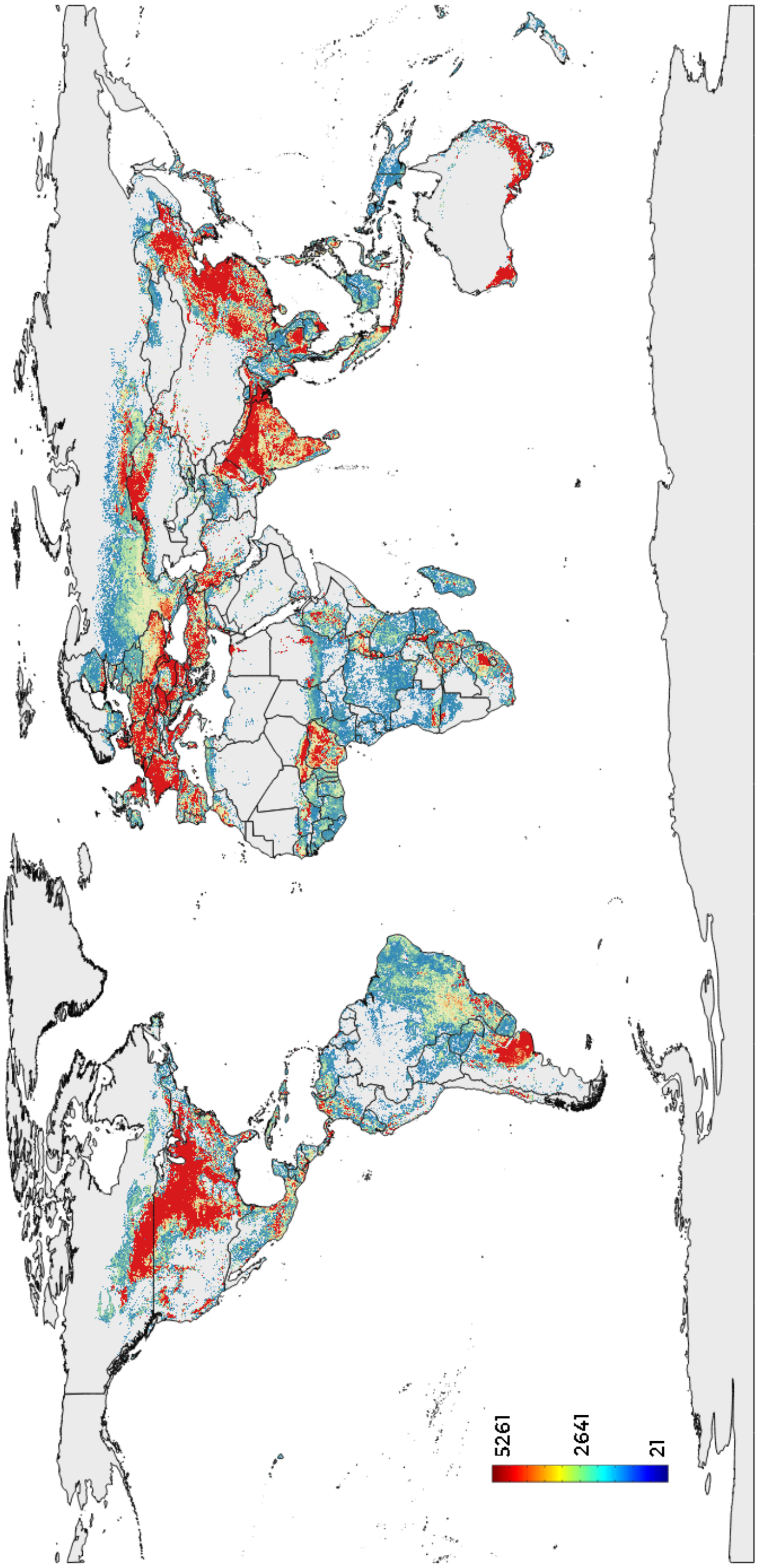
Minimize m^3_{water} utilized, maximizing gr_{prot} production

Crop distribution



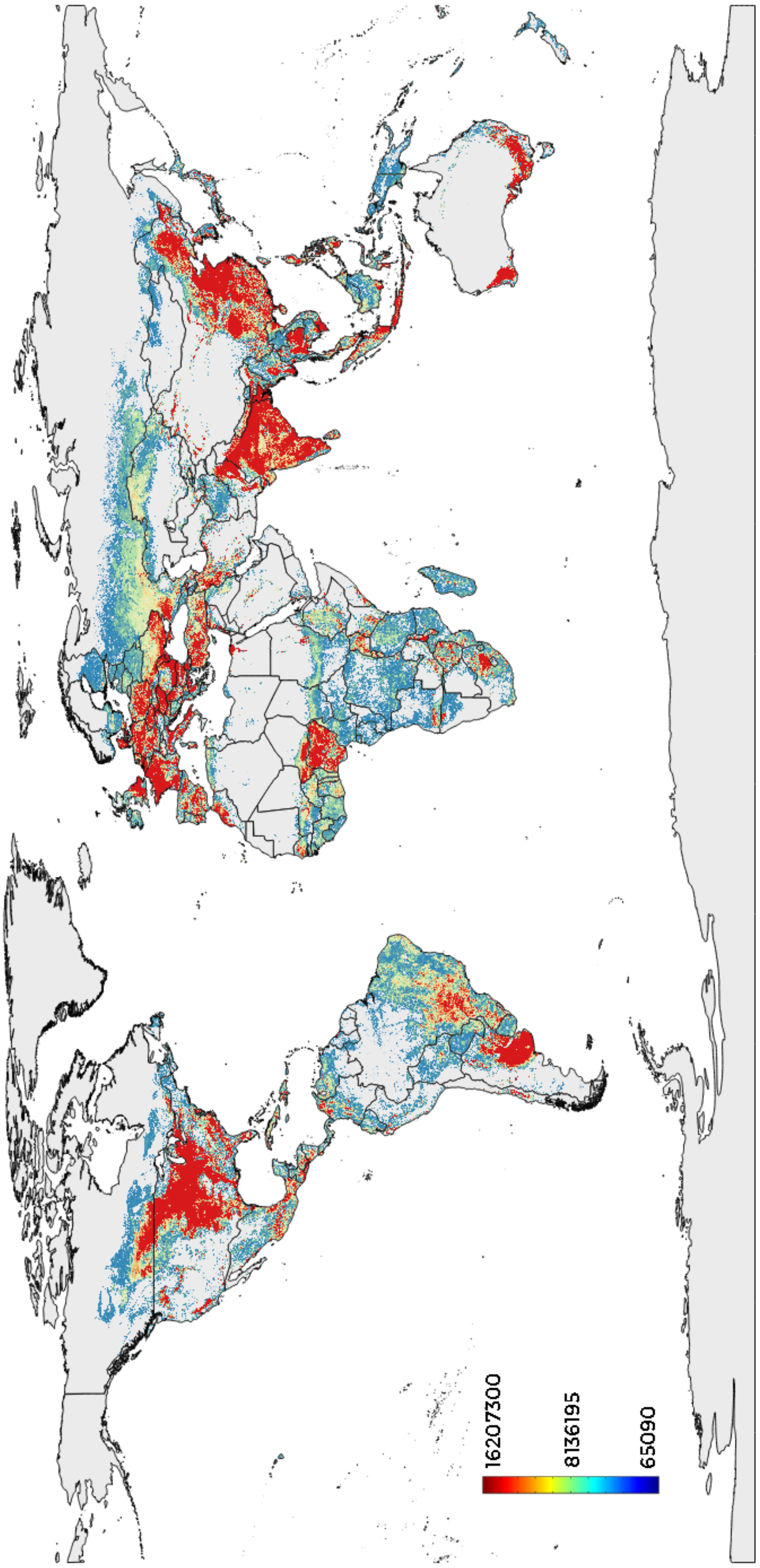
Minimize mwater3 utilized, maximizing grprot production

ton produced



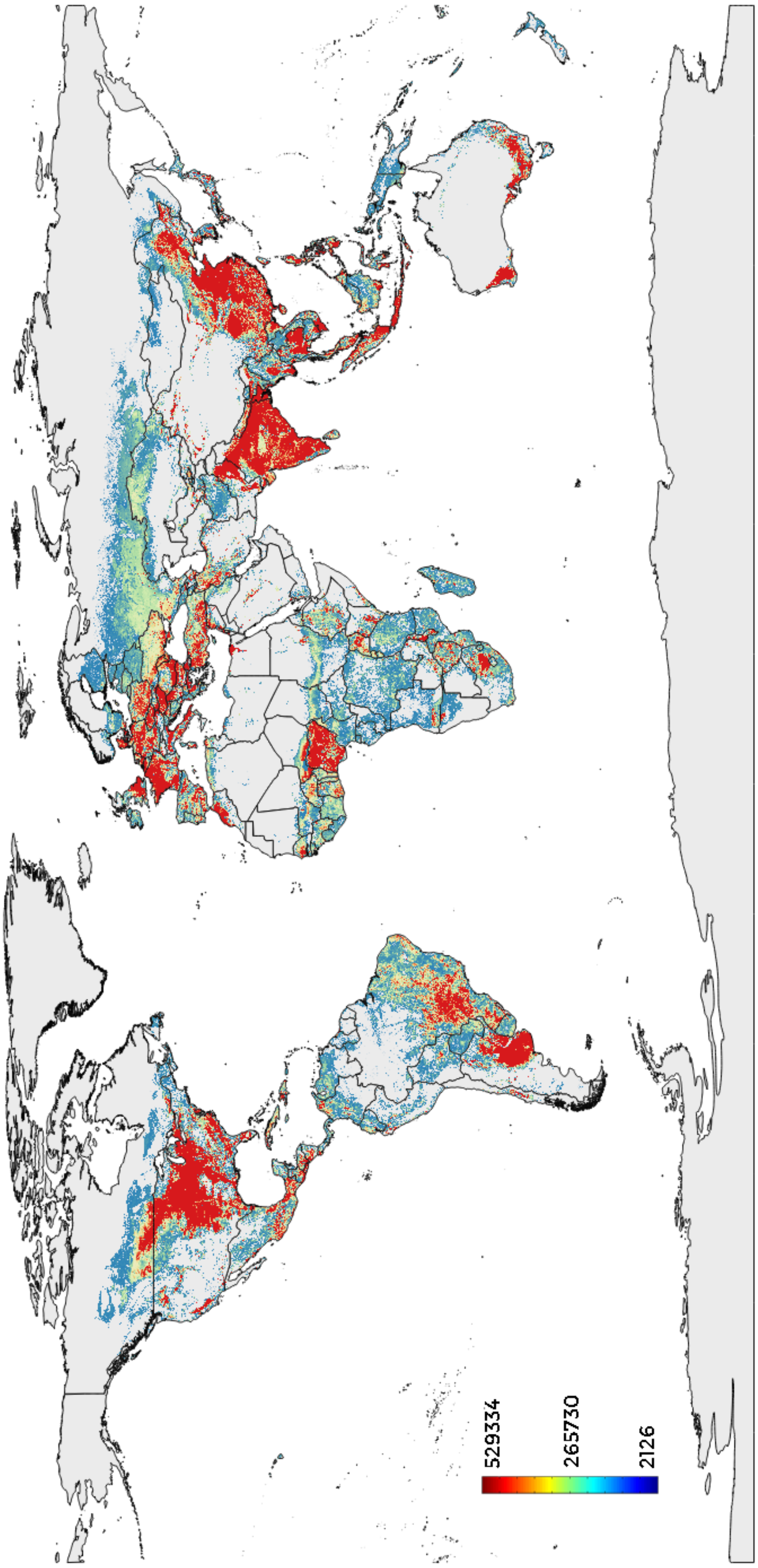
Minimize mwater3 utilized, maximizing grprot production

Gcal ***produced***



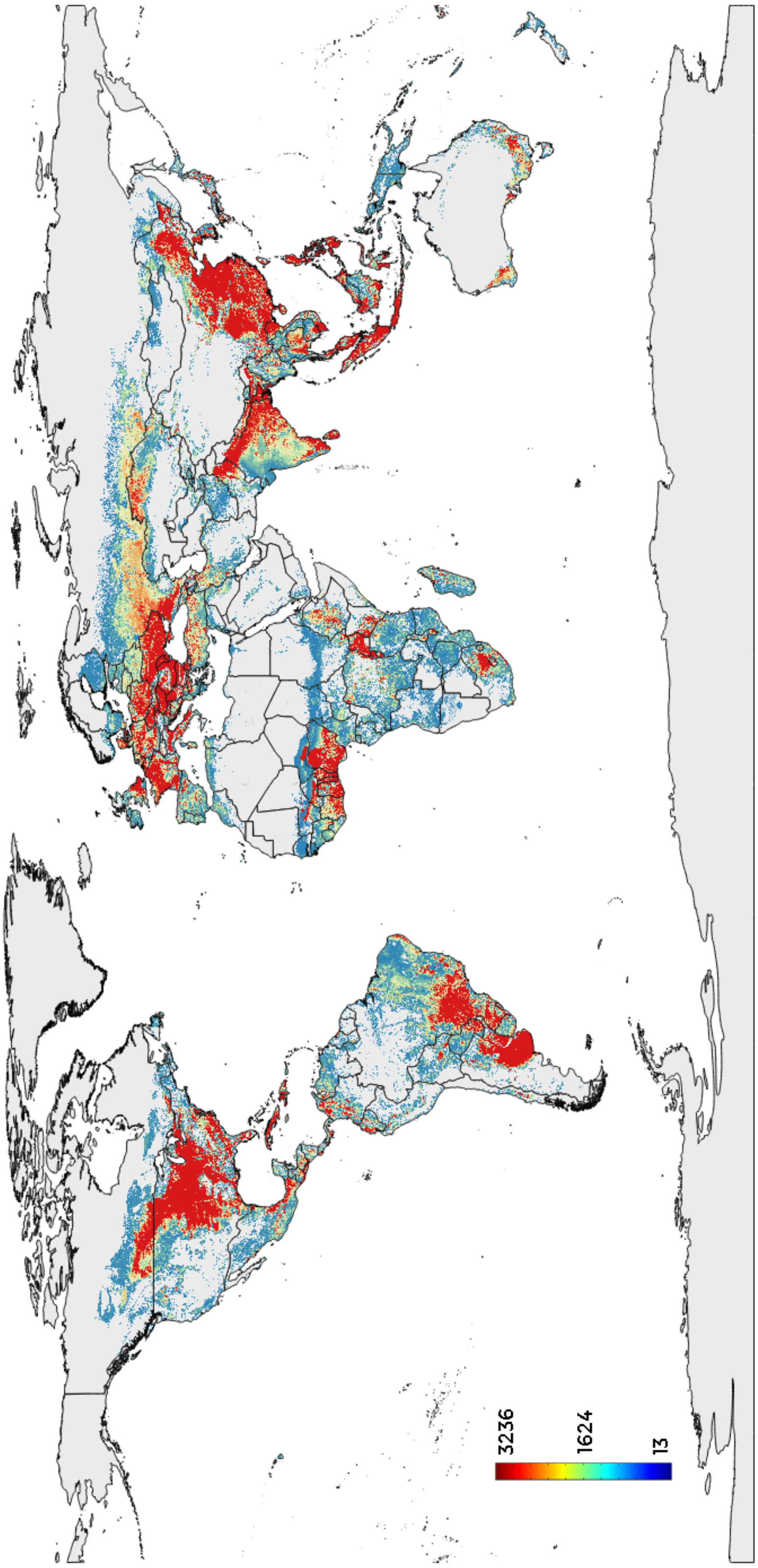
Minimize mwater3 utilized, maximizing grprot production

kg_{prot} ***produced***



Minimize mwater3 utilized, maximizing grprot production

m^3_{water} ***used***

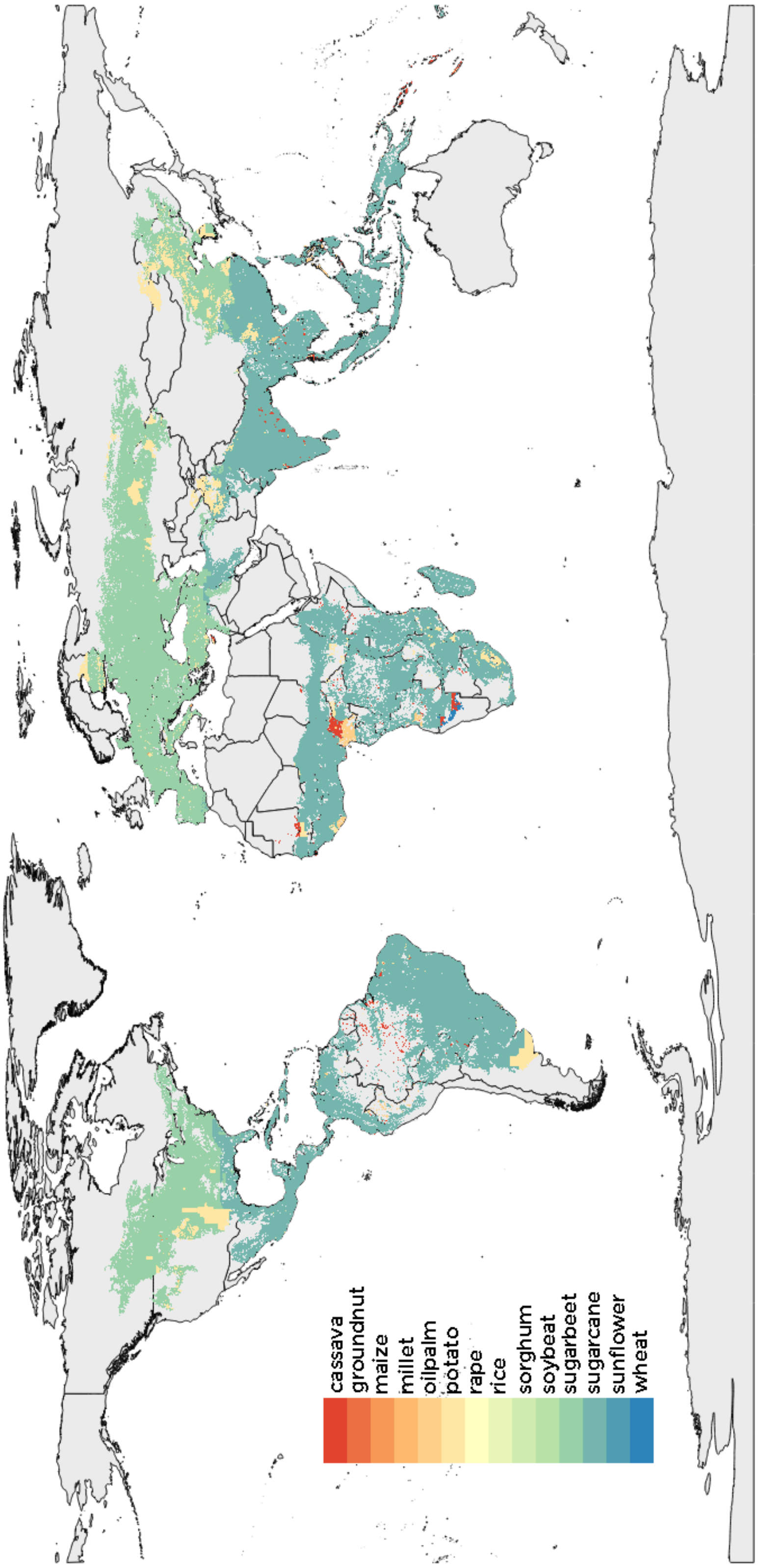


Appendix 3 – Maps soybean considered

This section will report the maps with results considering soybean. So the crop distribution is the same as mentioned above, but if within a cell we have soybean planted, we won't subtract that land to soybean and we'll keep it, since soybean it's used to regenerate the soil.

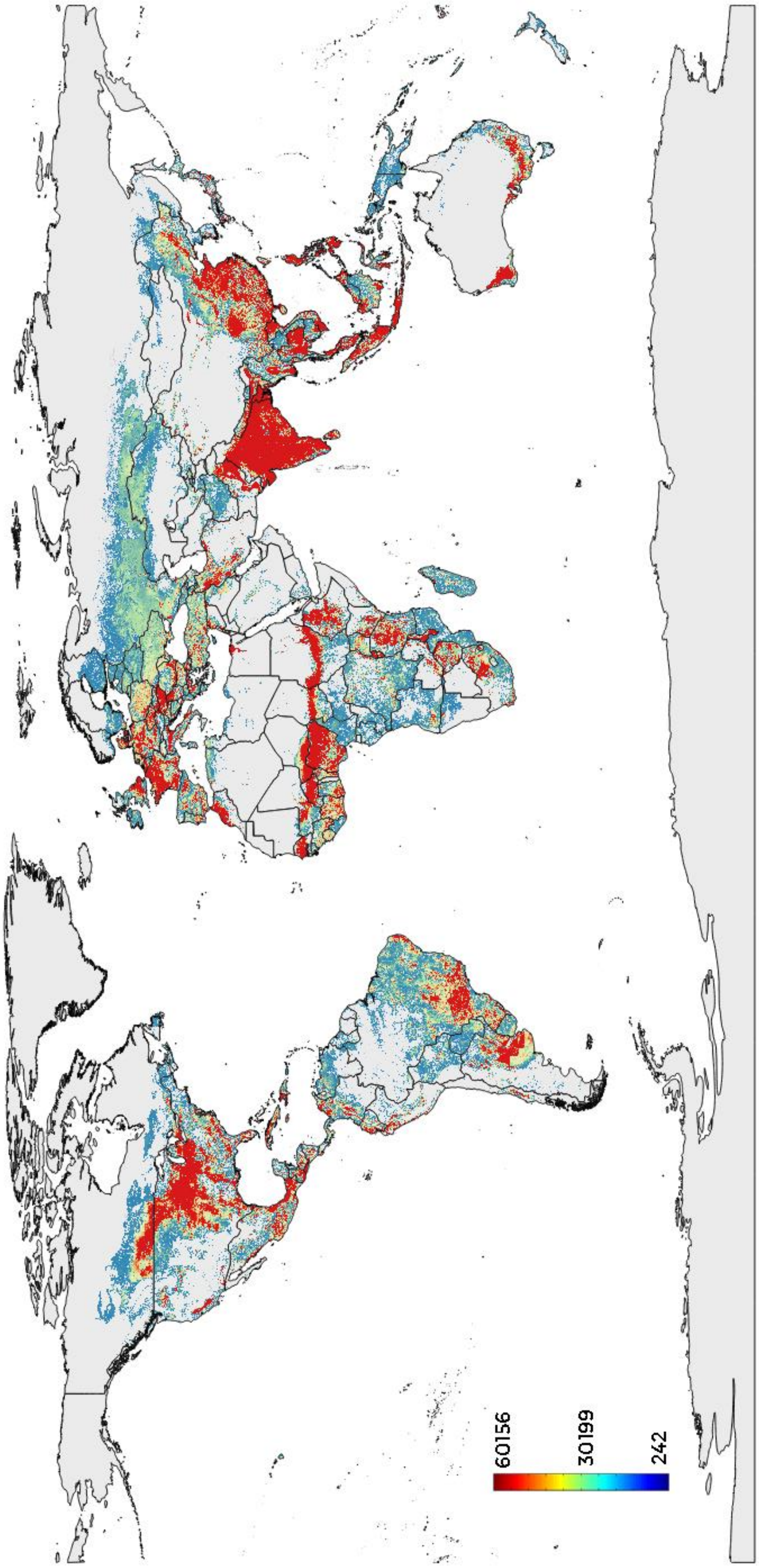
Maximize kg production

Crop distribution

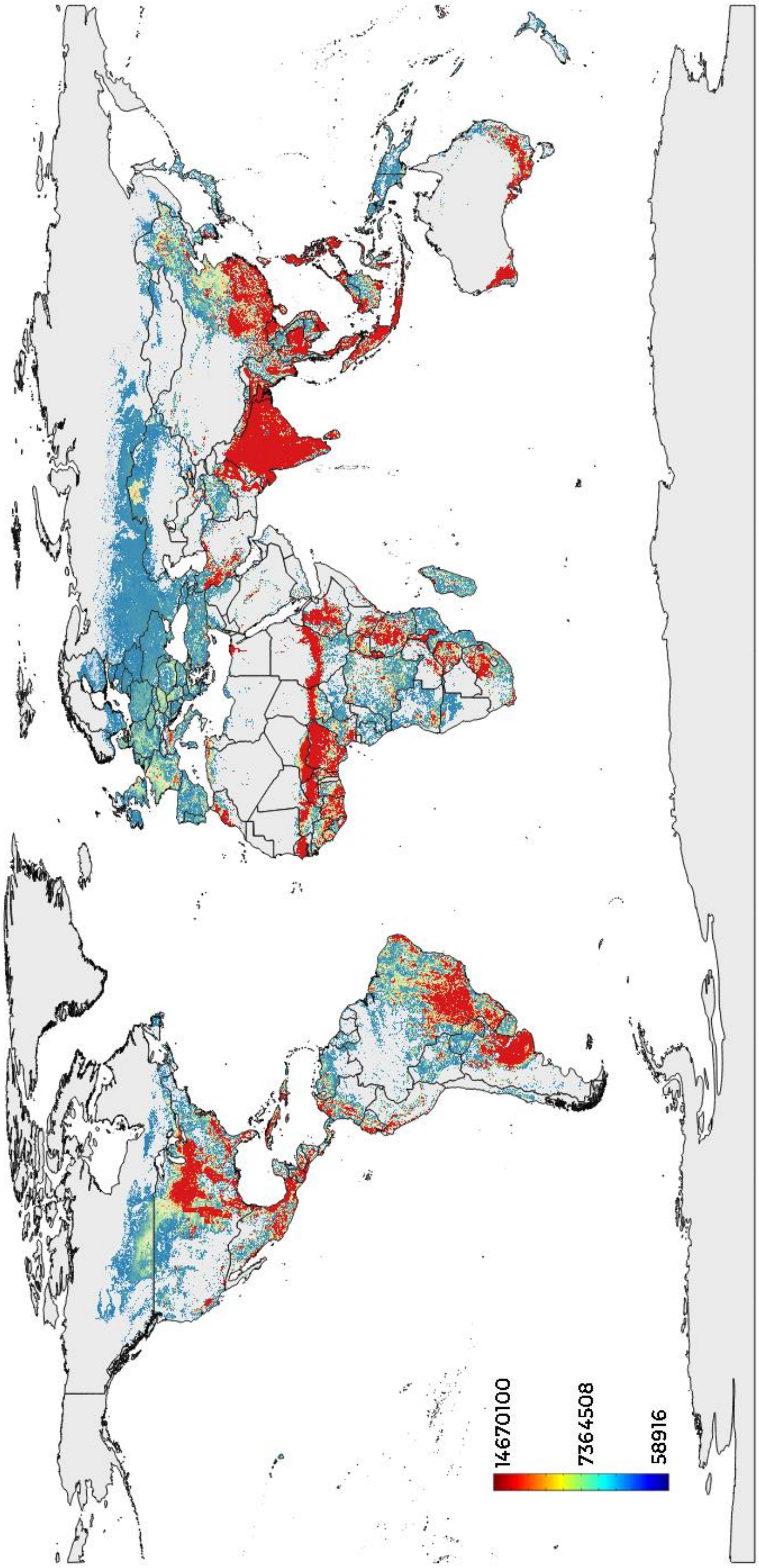


Maximize kg production

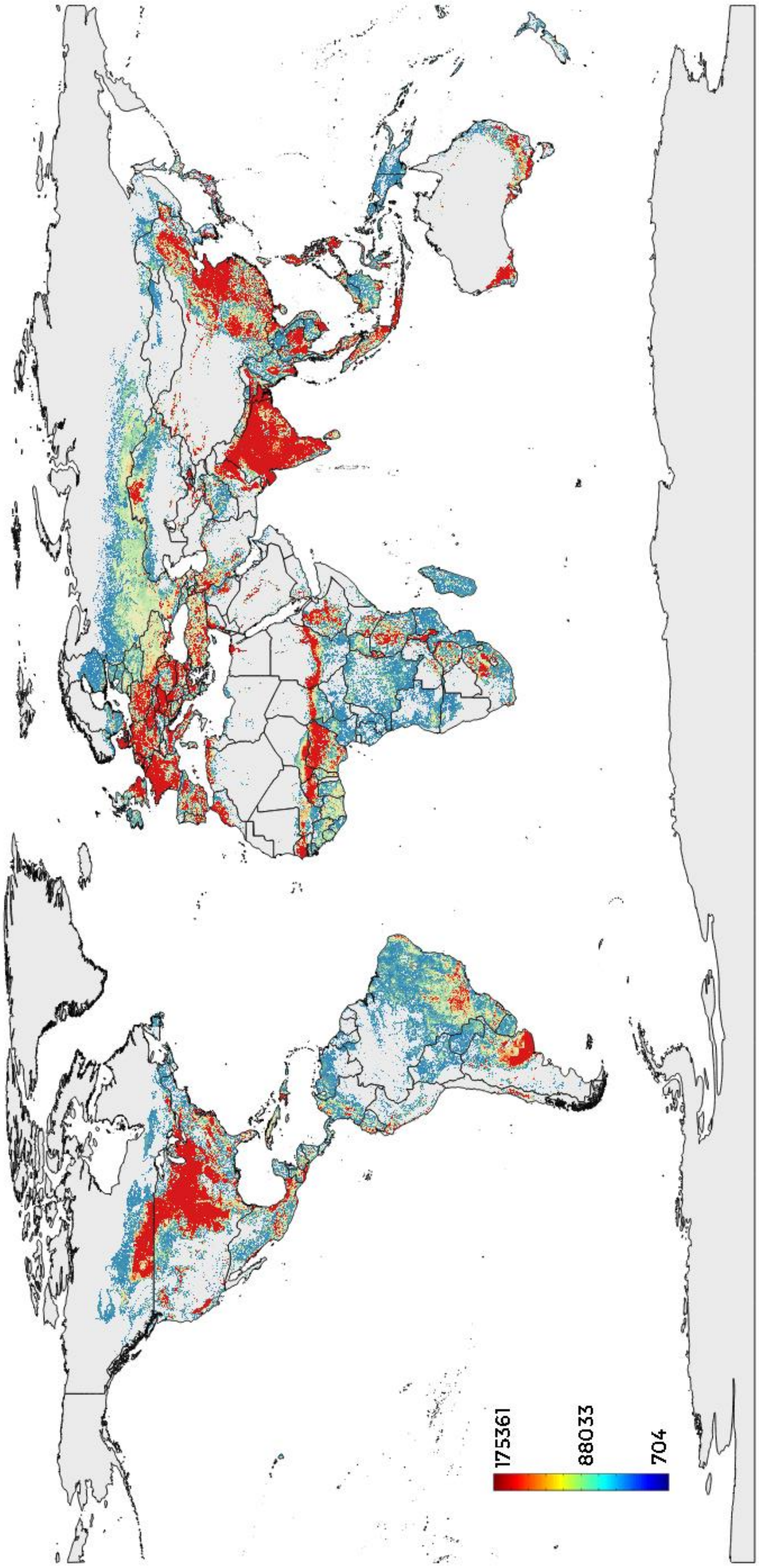
ton *produced*



$Gcal$ ***produced***

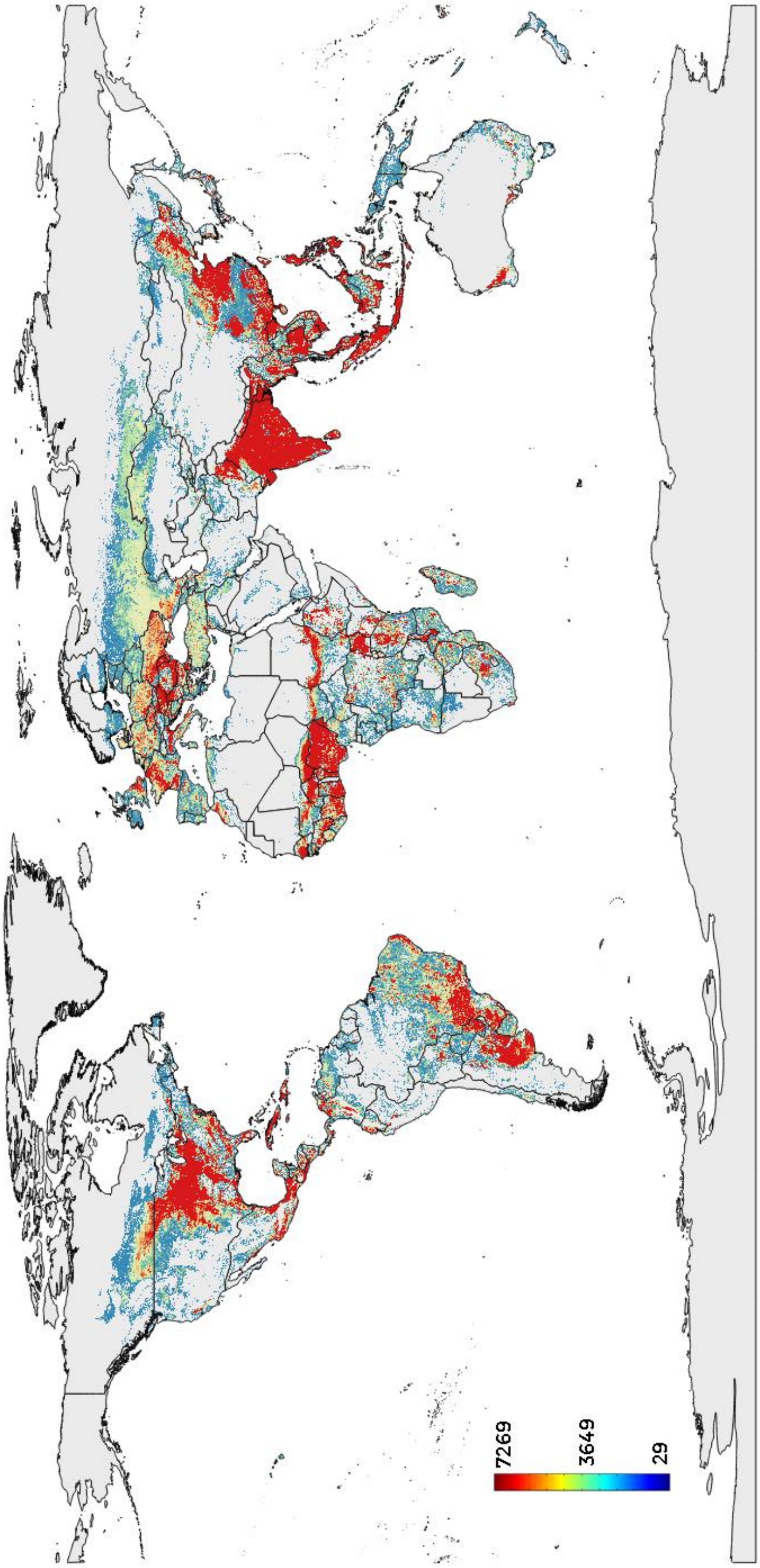


kg_{prot} produced



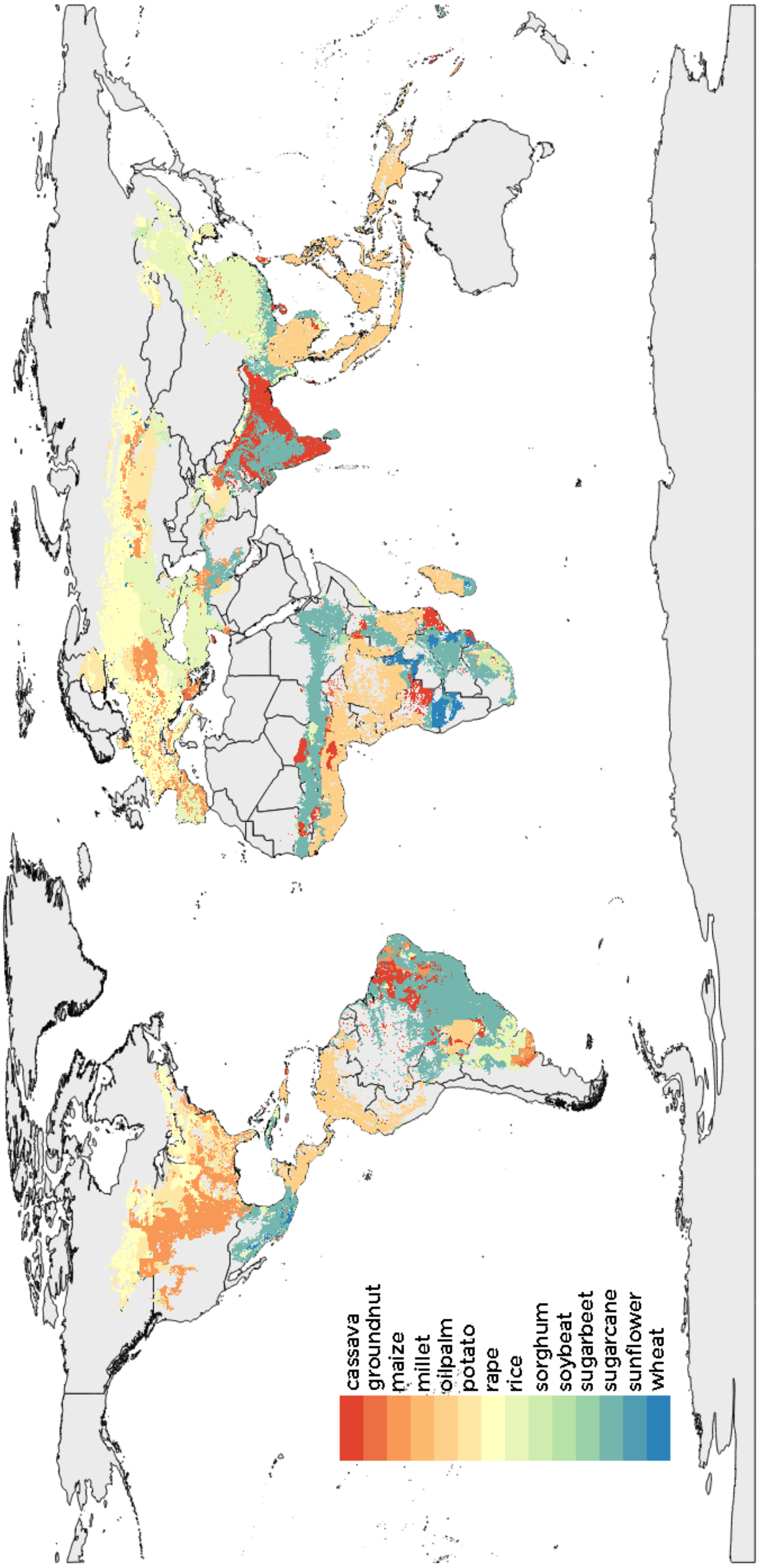
Maximize kg production

m^3_{water} ***used***

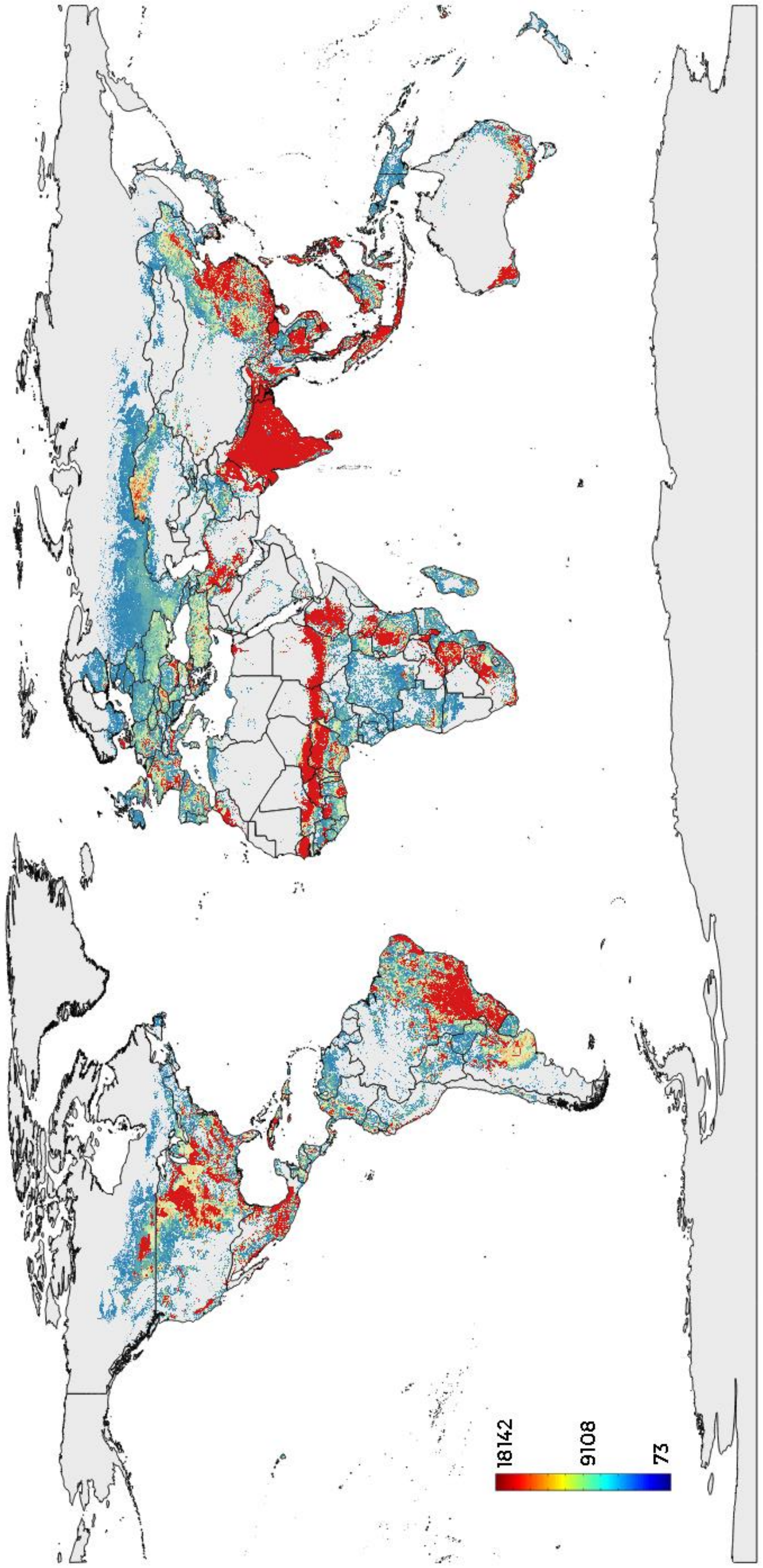


Maximize $kcal$ production

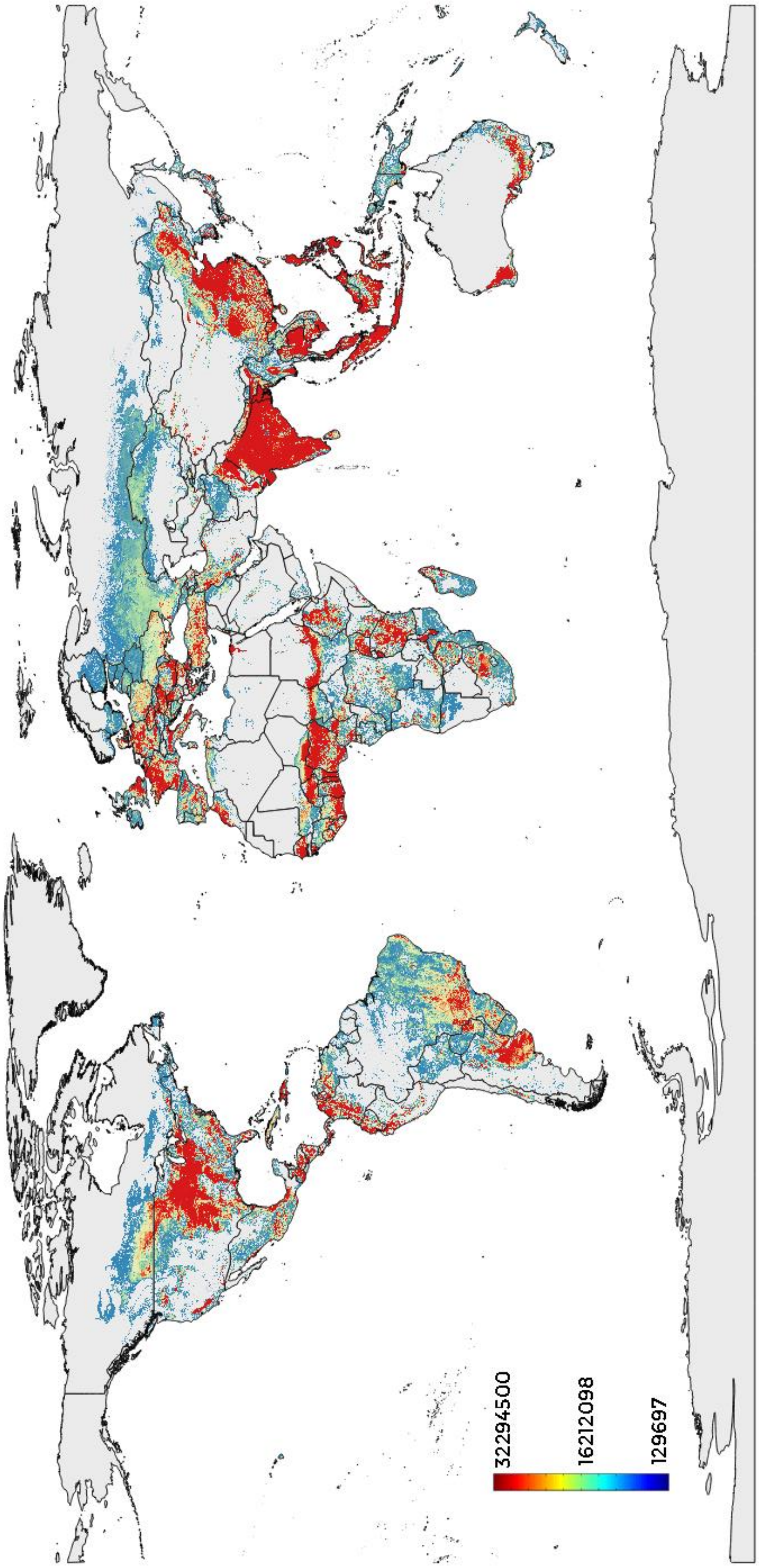
Crop distribution



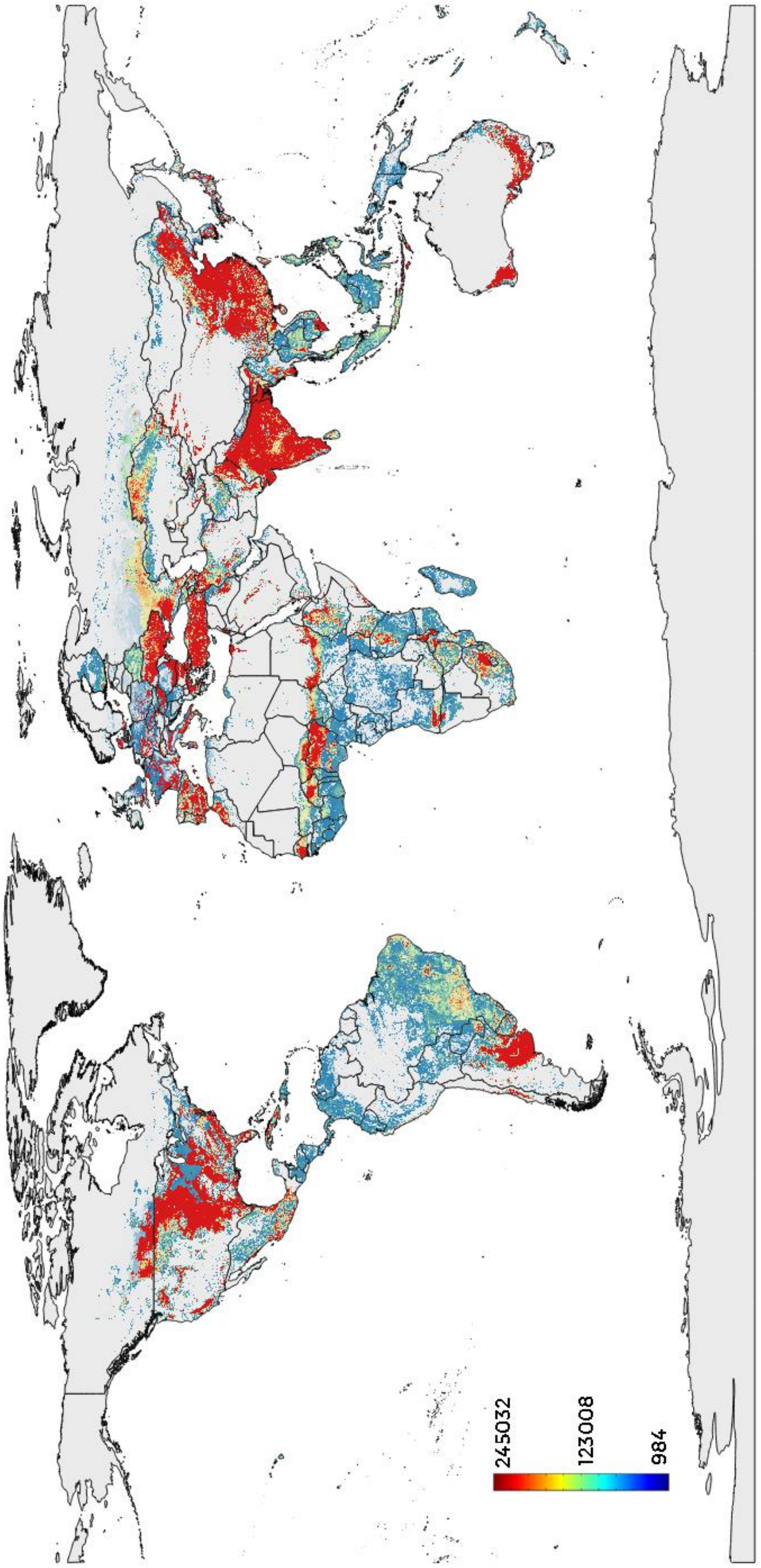
ton ***produced***



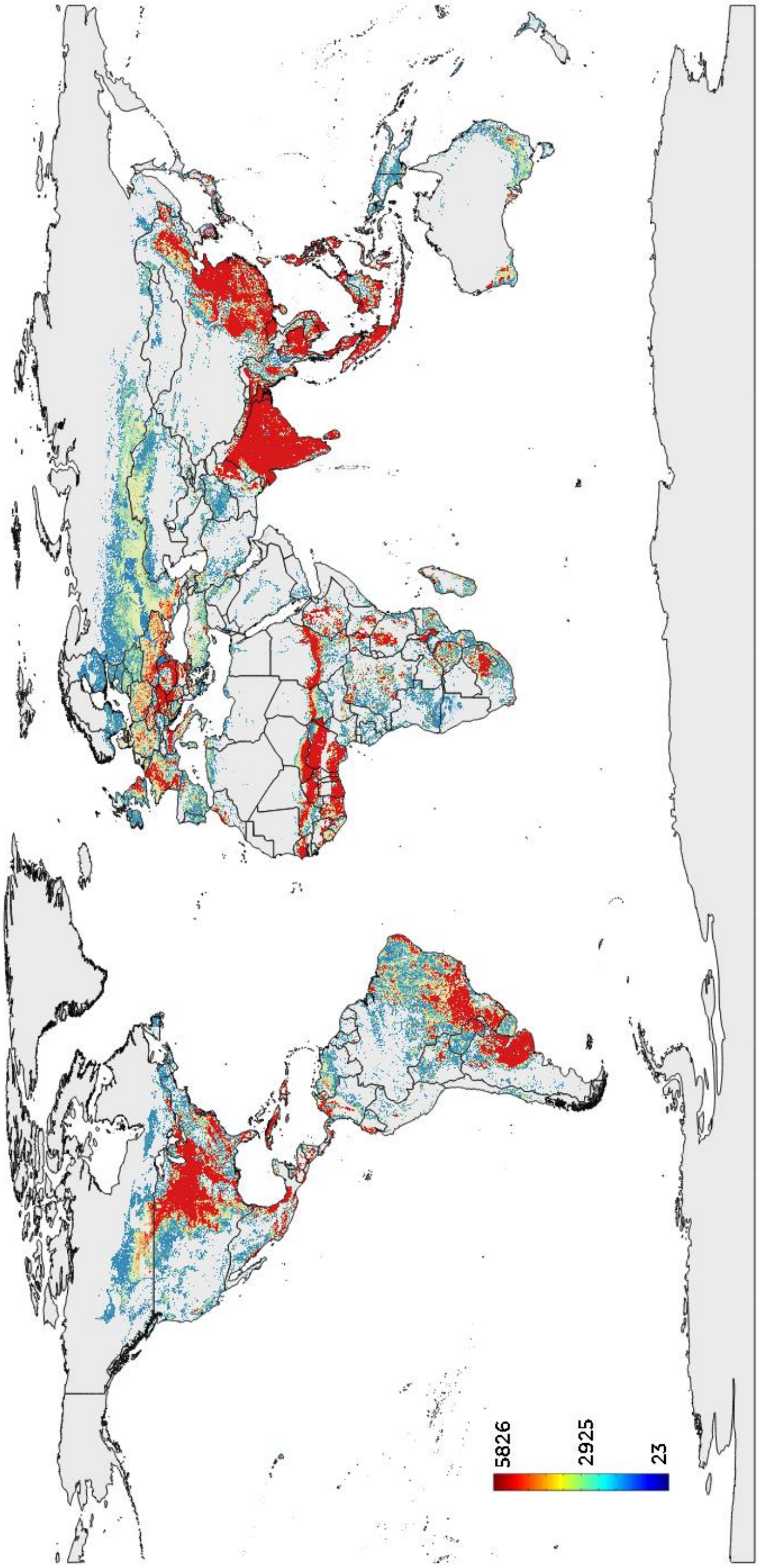
$Gcal$ ***produced***



kg_{prot} ***produced***

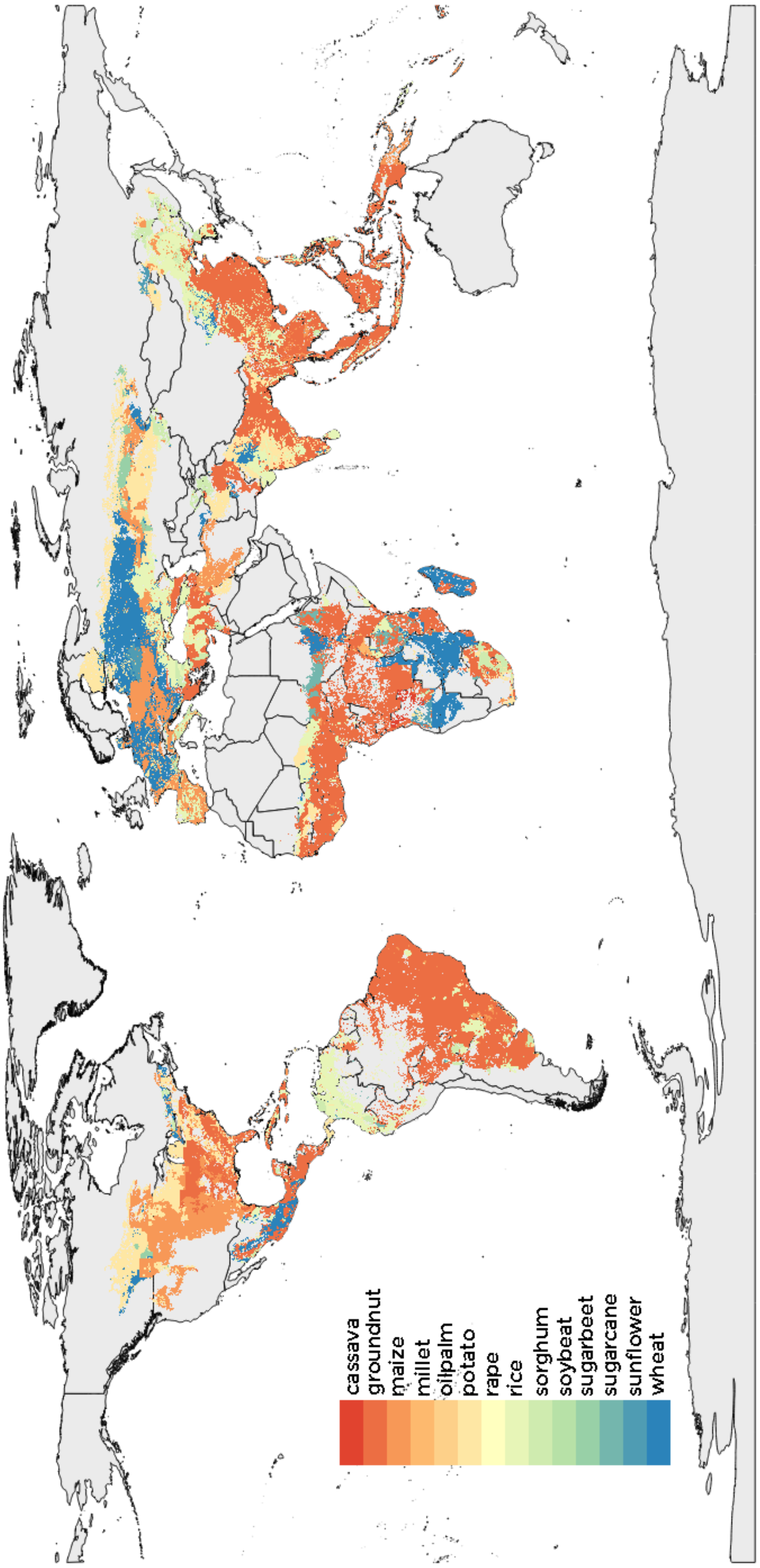


m^3_{water} ***used***



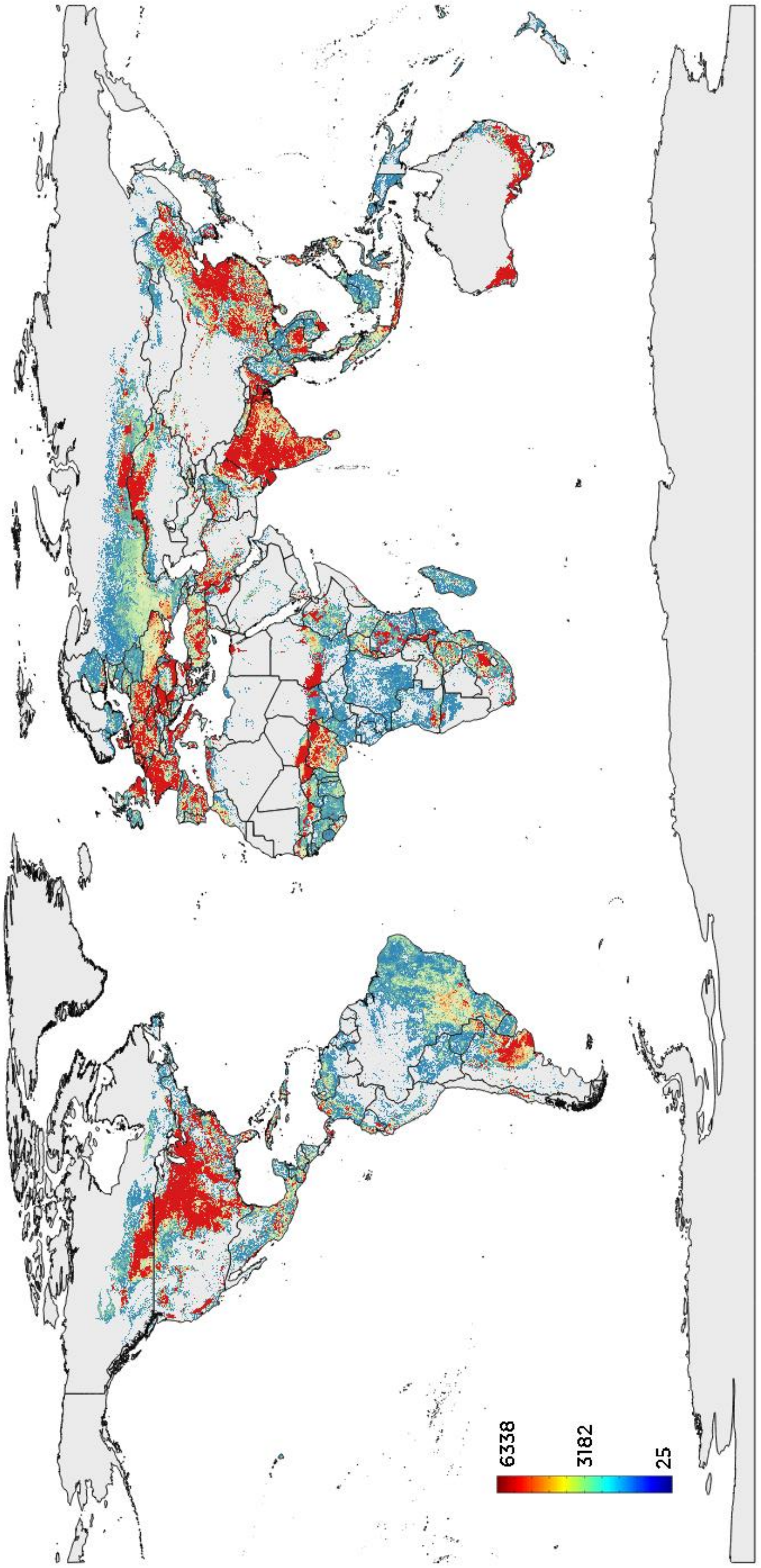
Maximize gr_{prot} production

Crop distribution

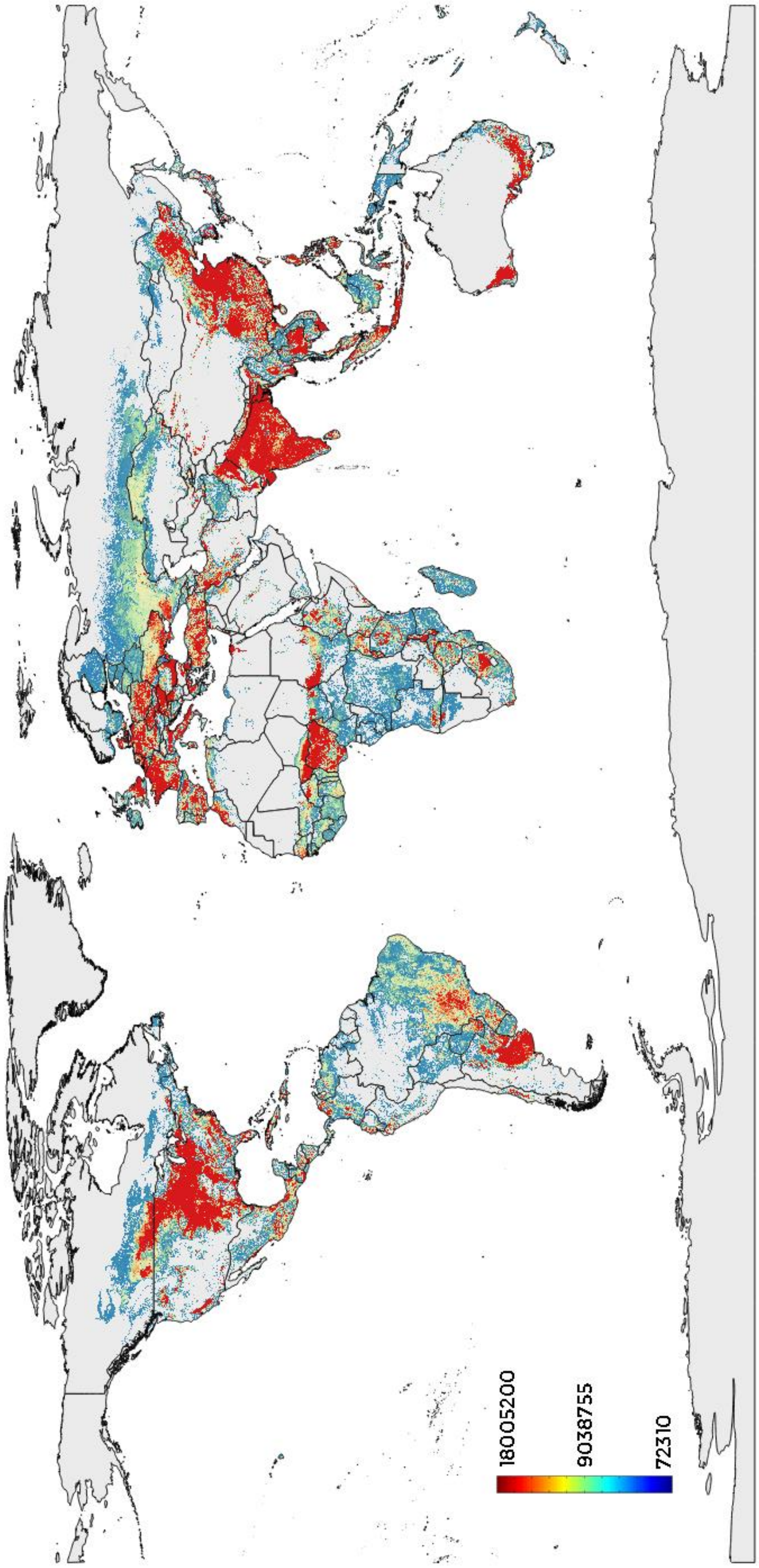


Maximize grprot production

ton produced

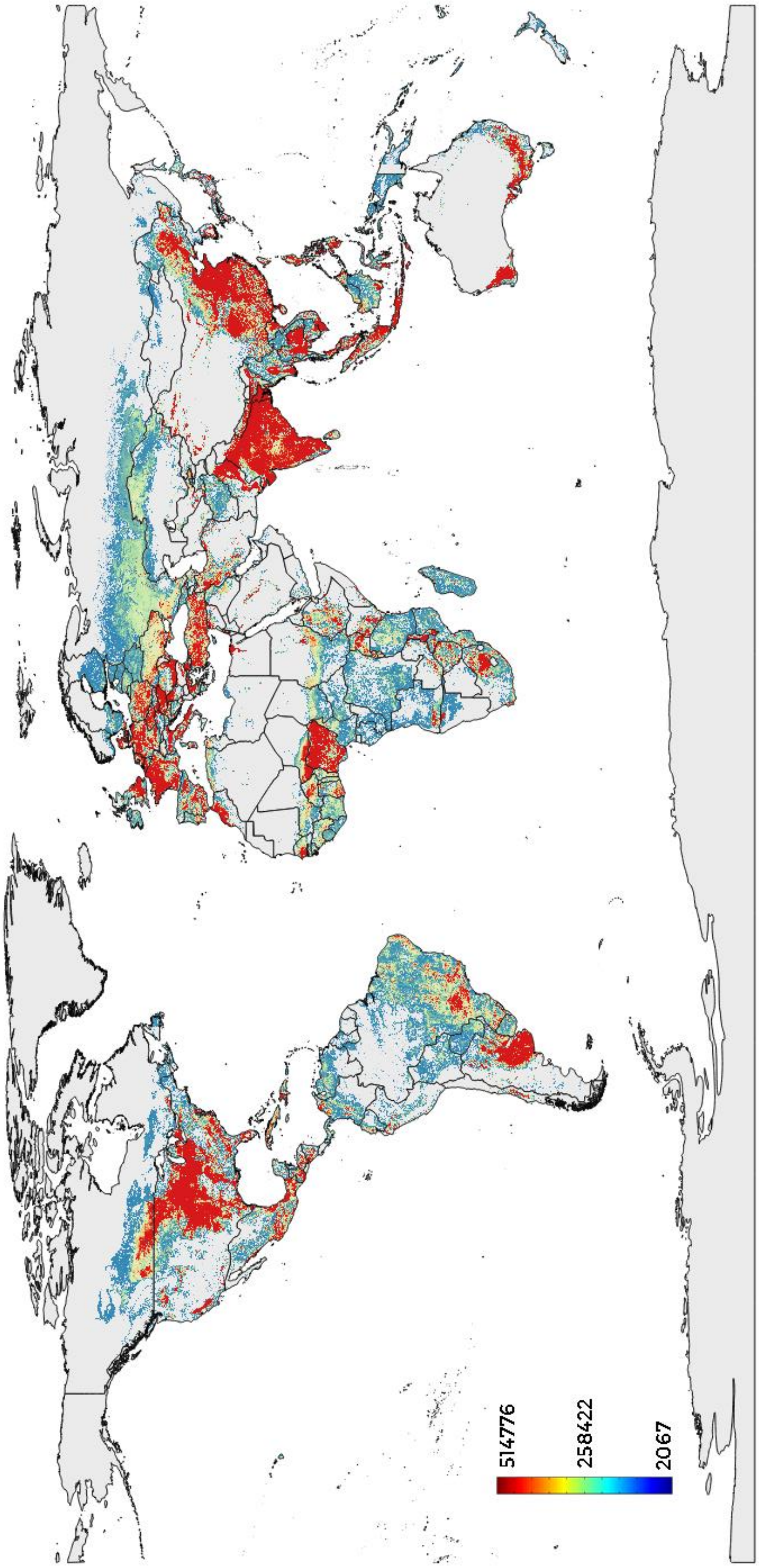


Gcal ***produced***

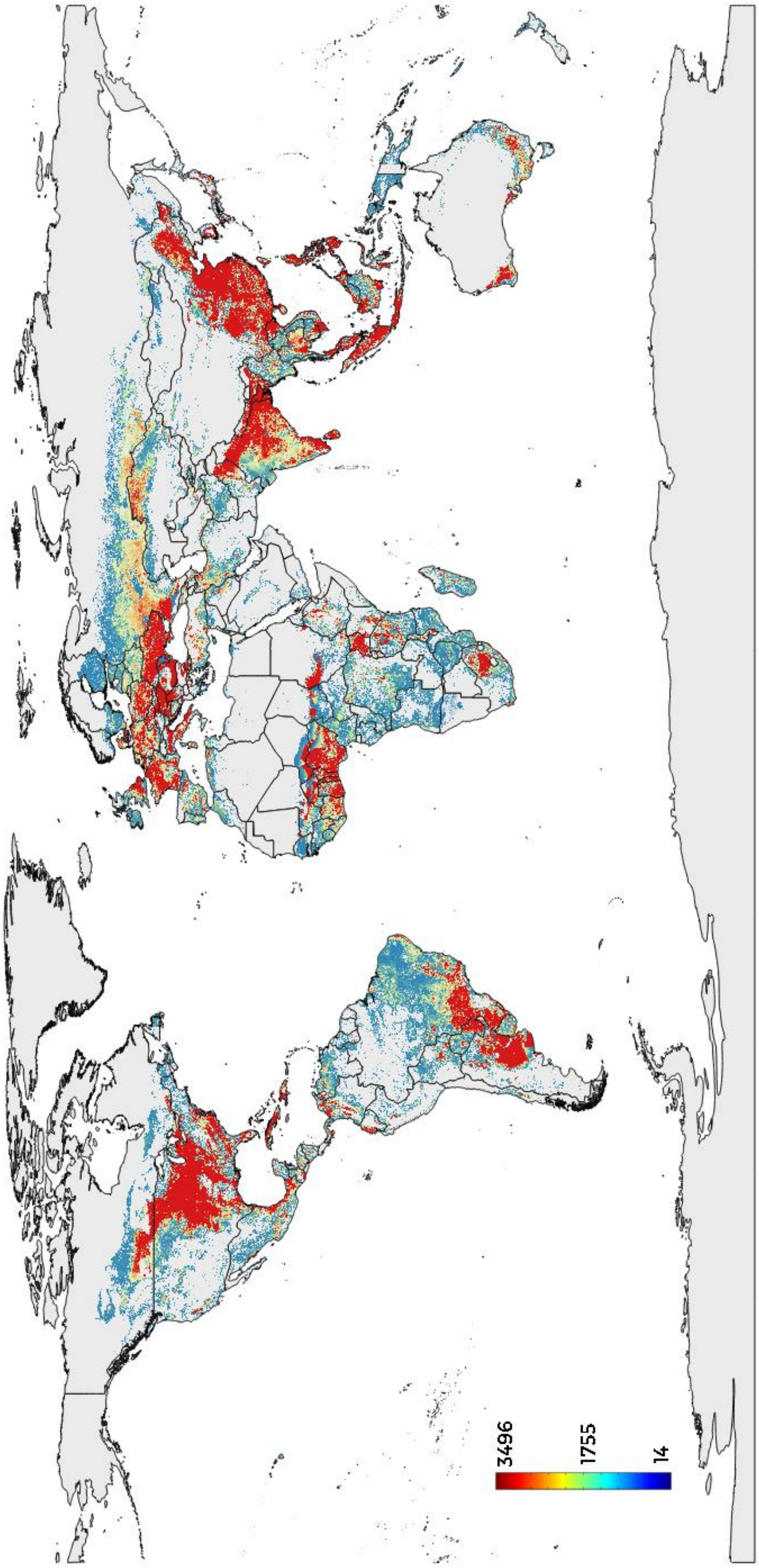


Maximize grprot production

kg_{prot} ***produced***

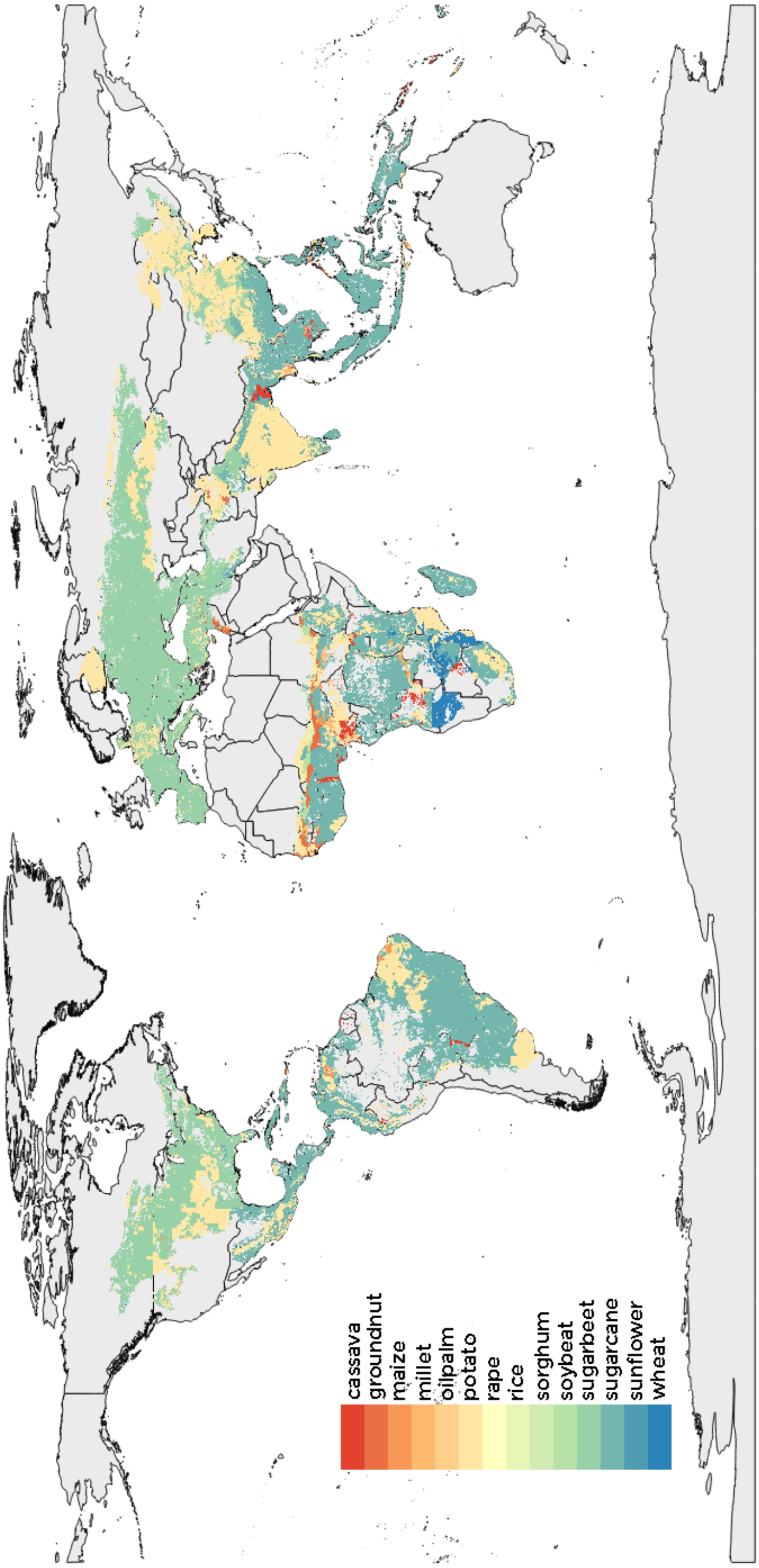


m^3_{water} ***used***

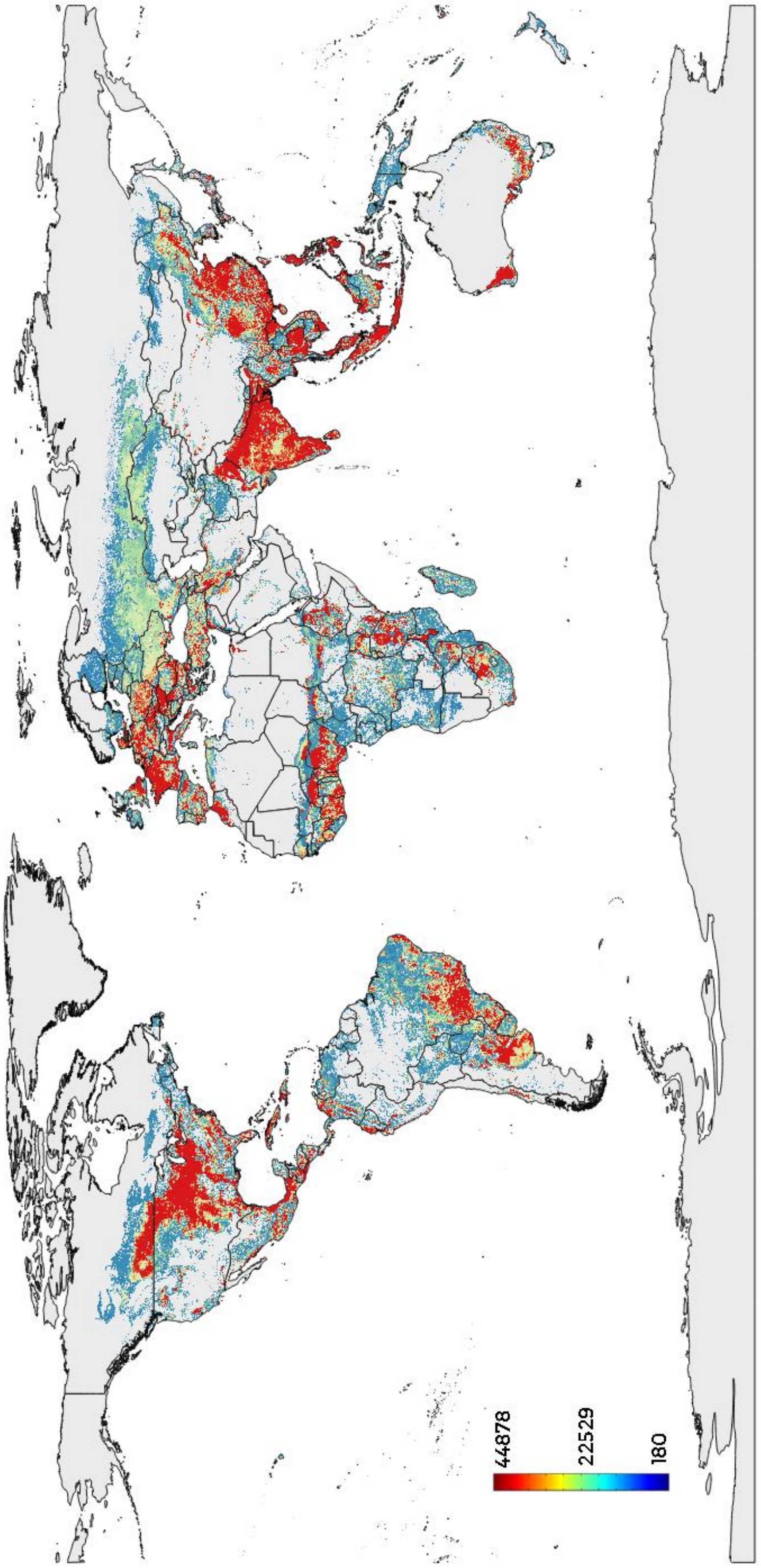


Minimize m_{water}^3 utilized, maximizing kg production

Crop distribution

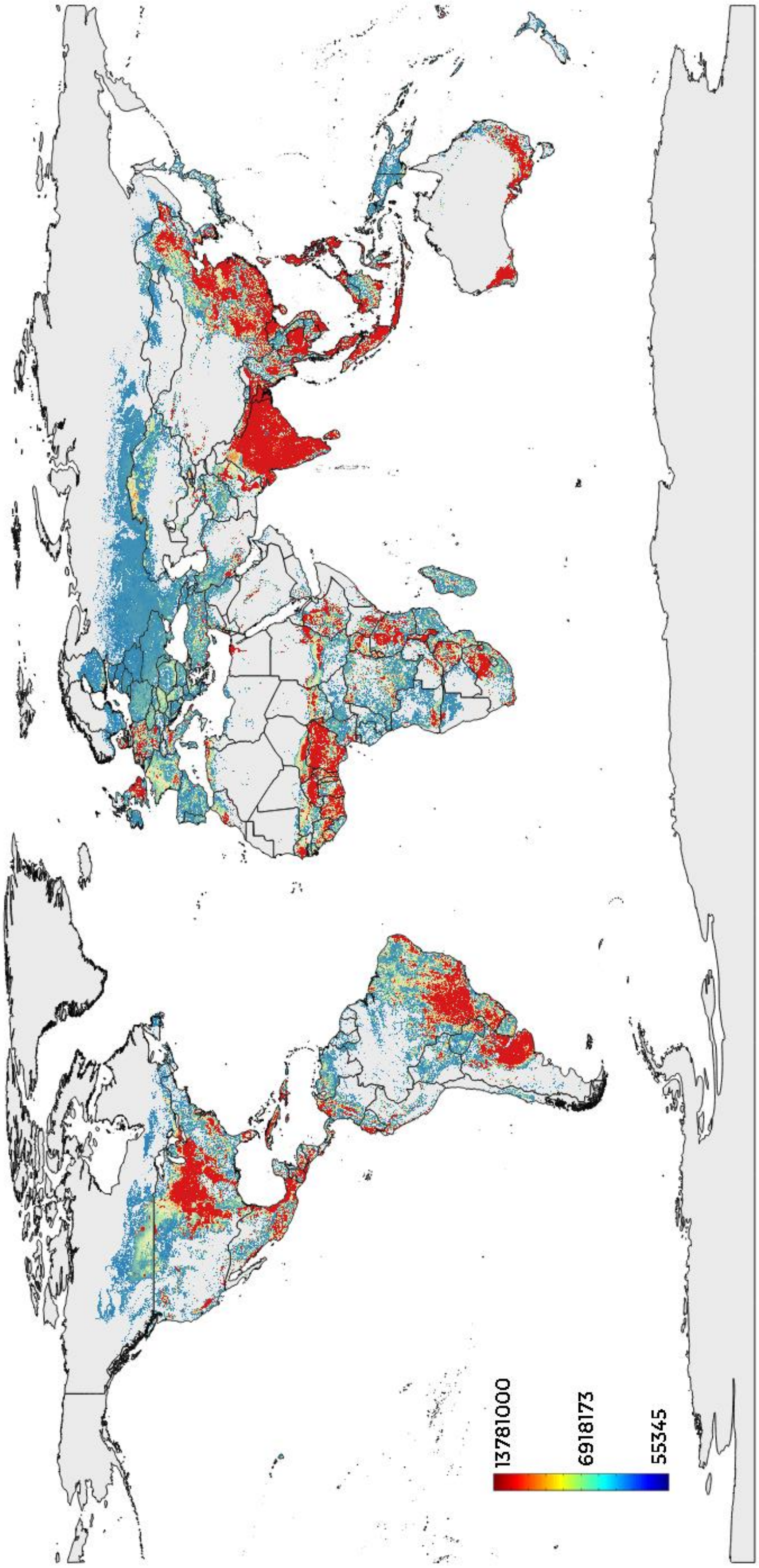


ton ***produced***

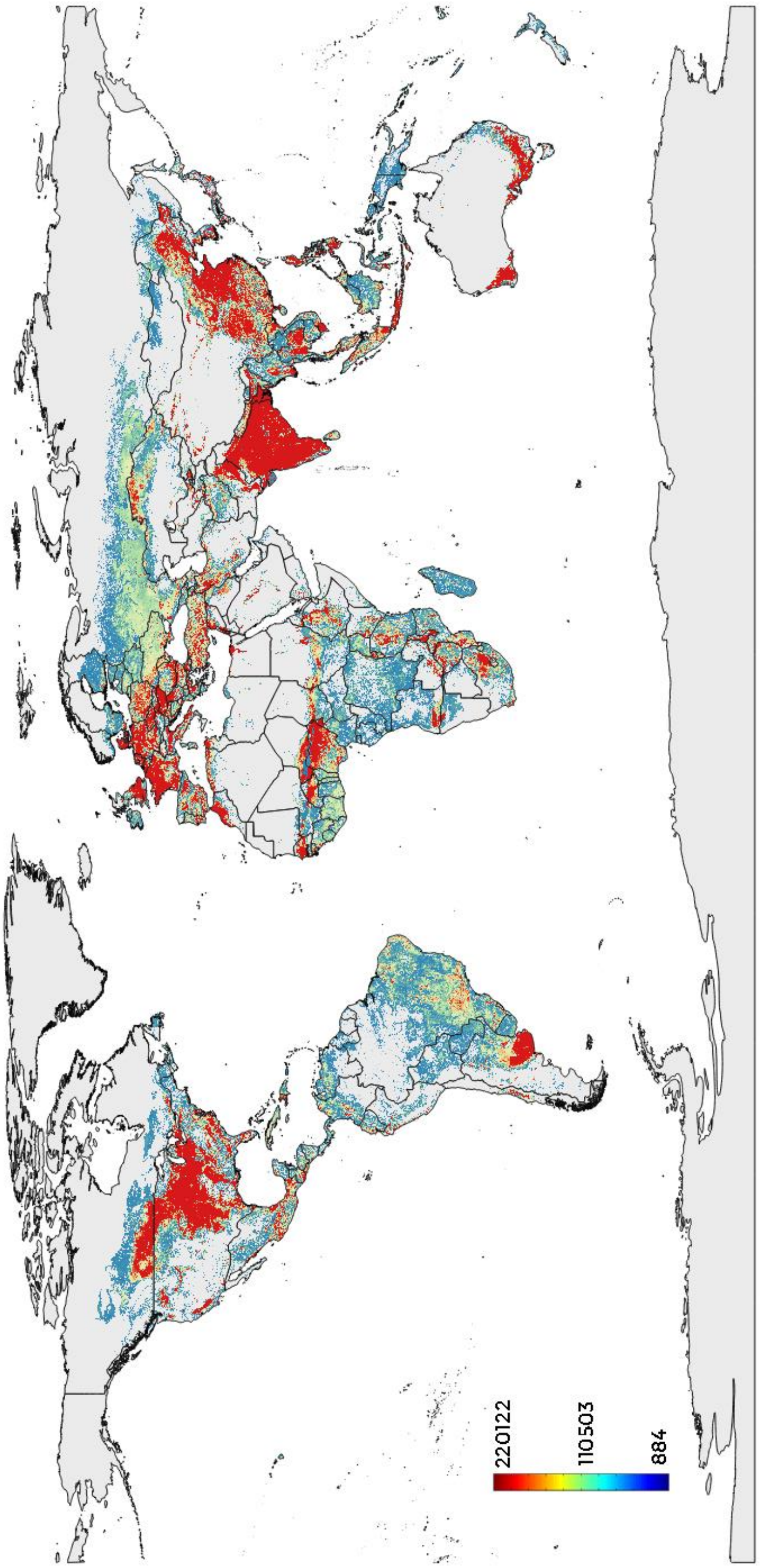


Minimize $mwater3$ utilized, maximizing kg production

$Gcal$ **produced**

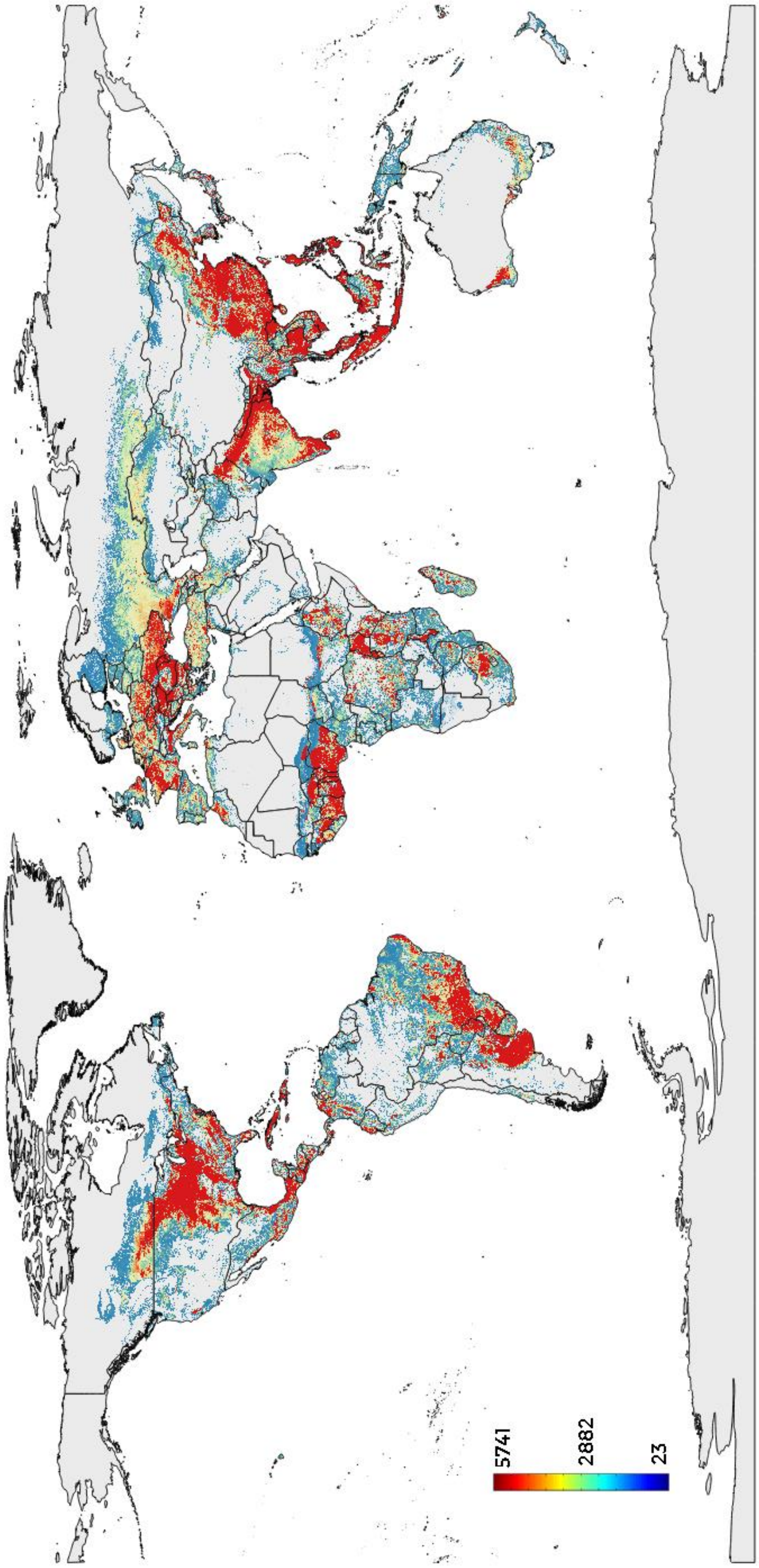


kg_{prot} ***produced***



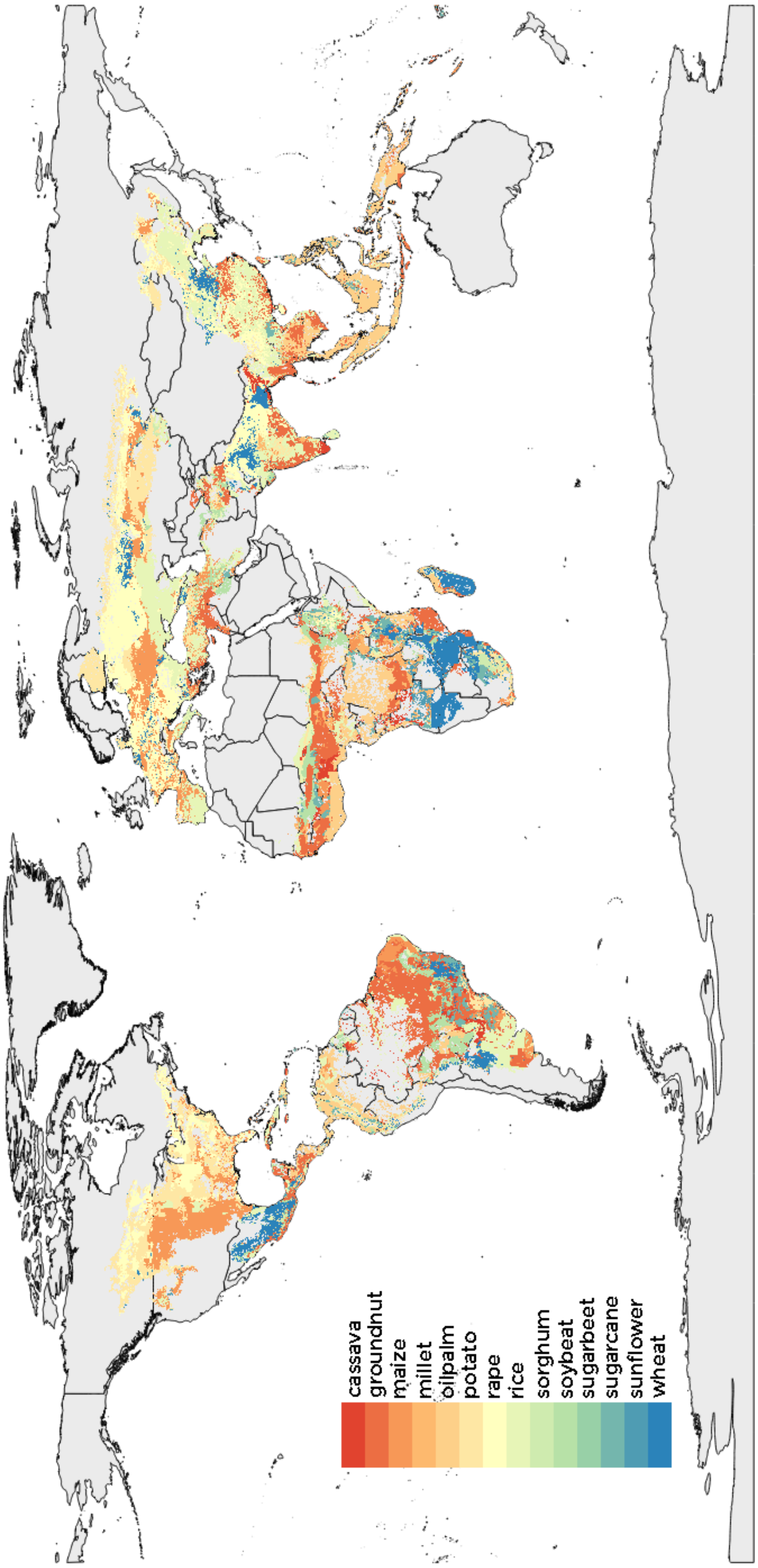
Minimize $mwater3$ utilized, maximizing kg production

m_{water}^{3} ***used***

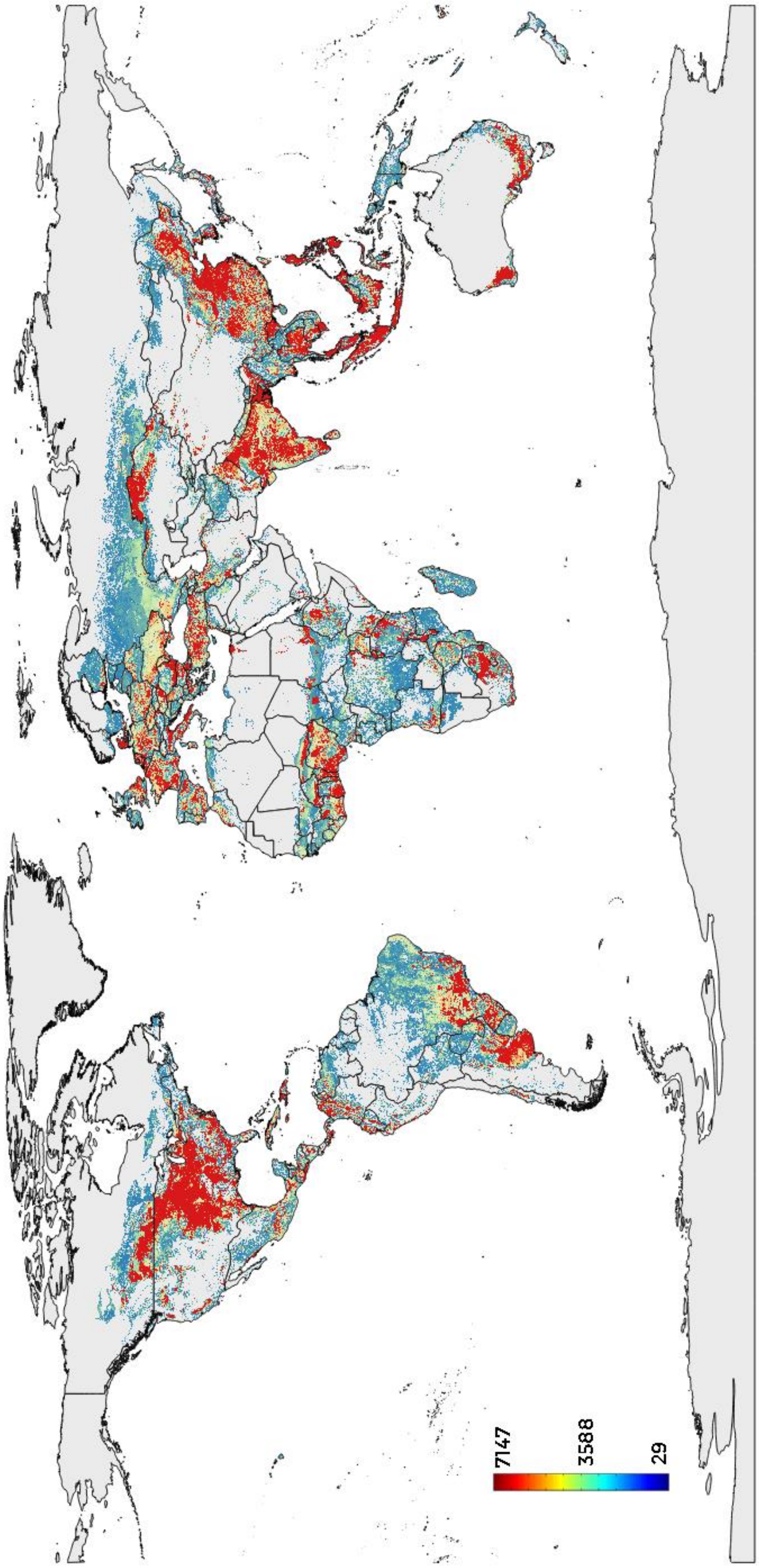


Minimize m_{water}^3 utilized, maximizing $kcal$ production

Crop distribution

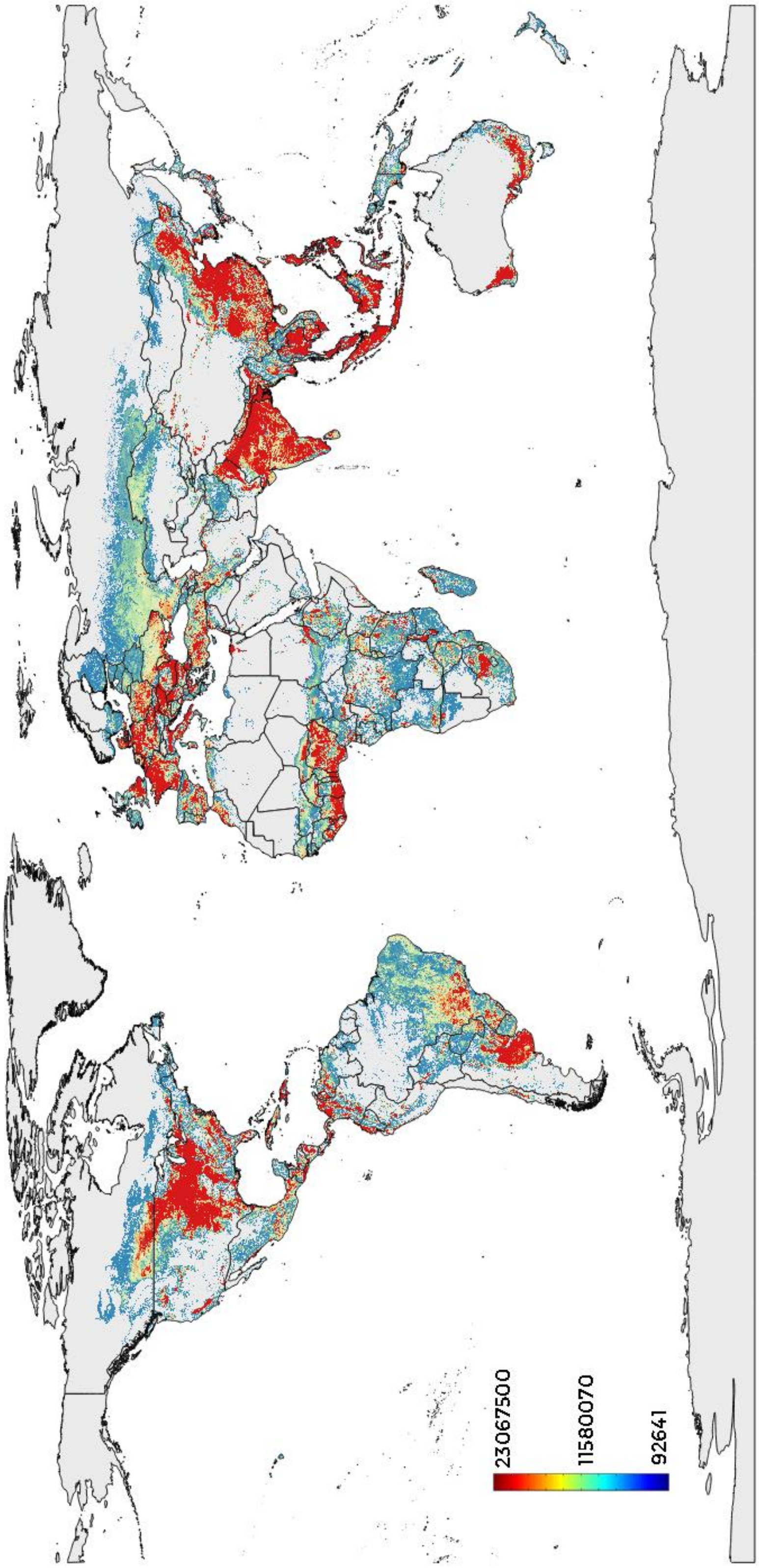


ton ***produced***

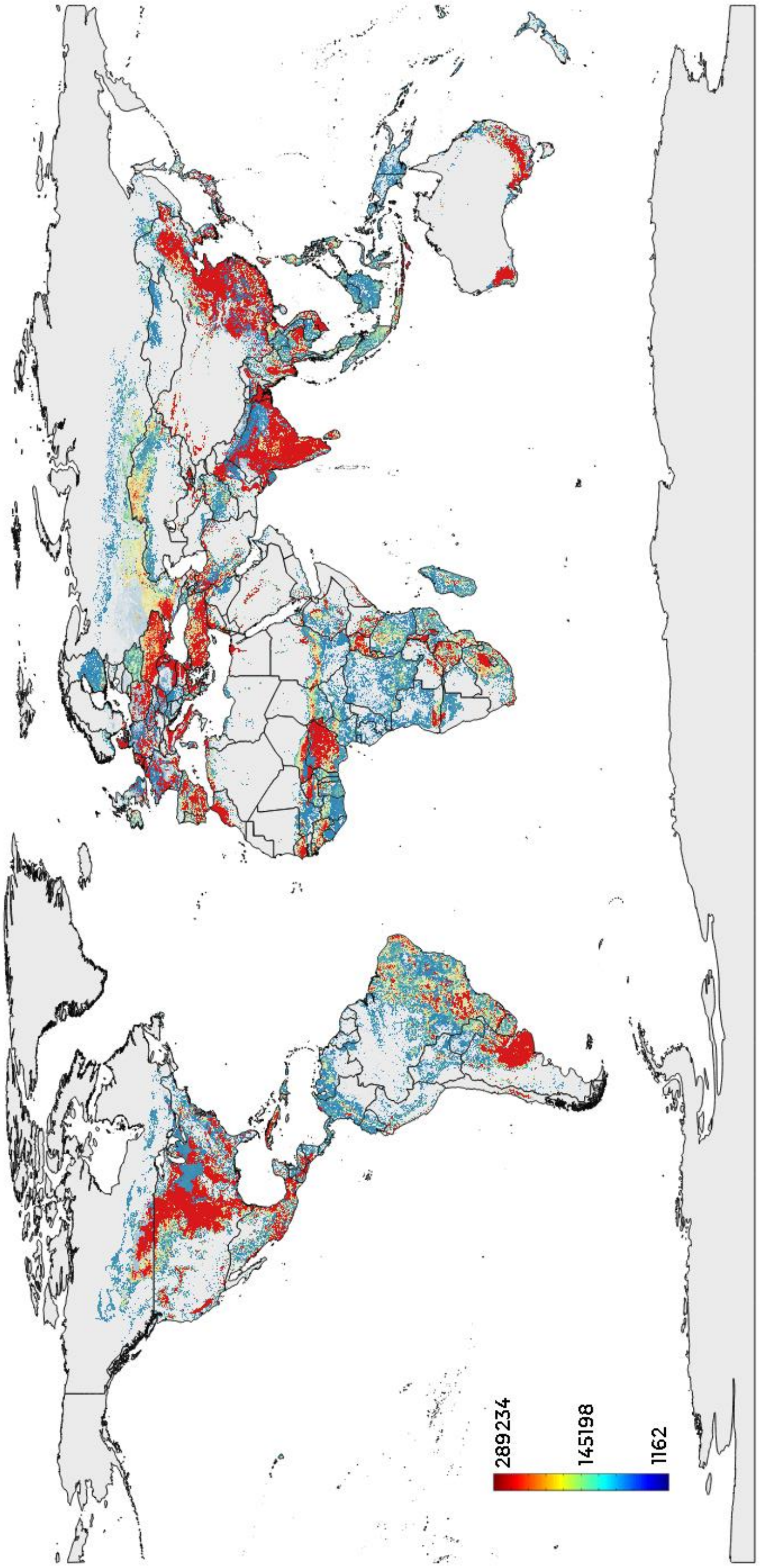


Minimize mwater3 utilized, maximizing kcal production

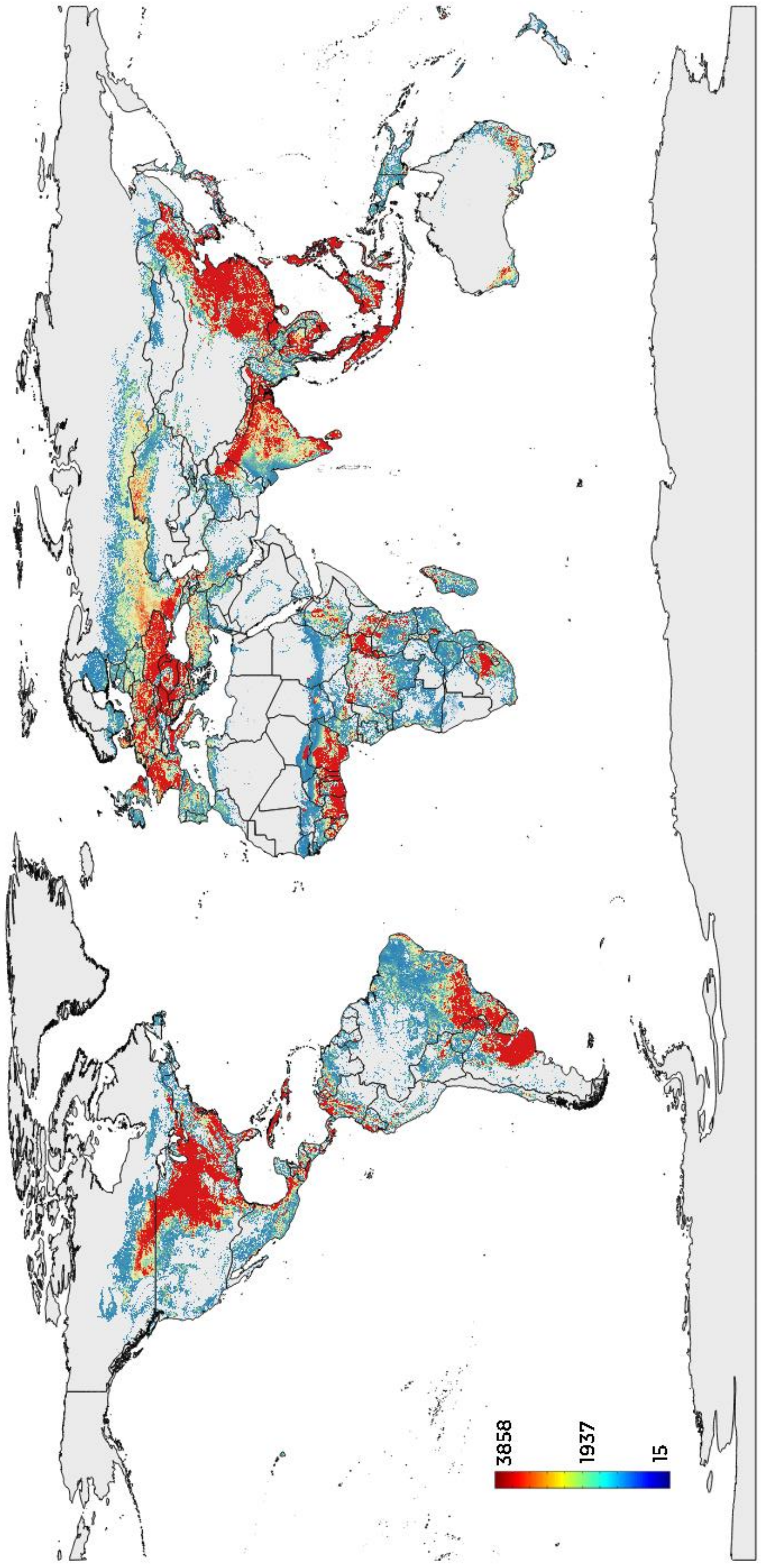
Gcal ***produced***



kg_{prot} ***produced***

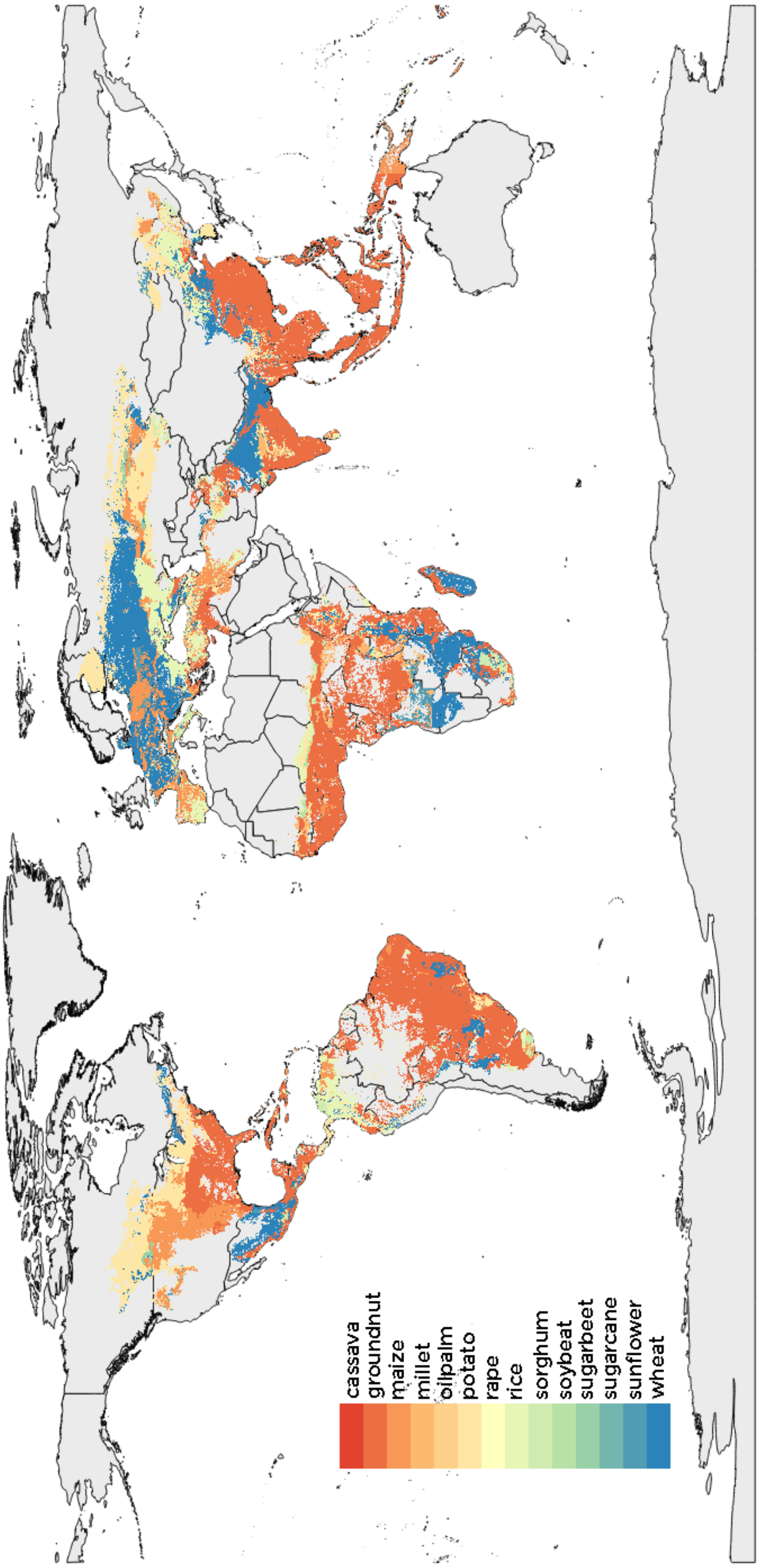


m^3_{water} ***used***



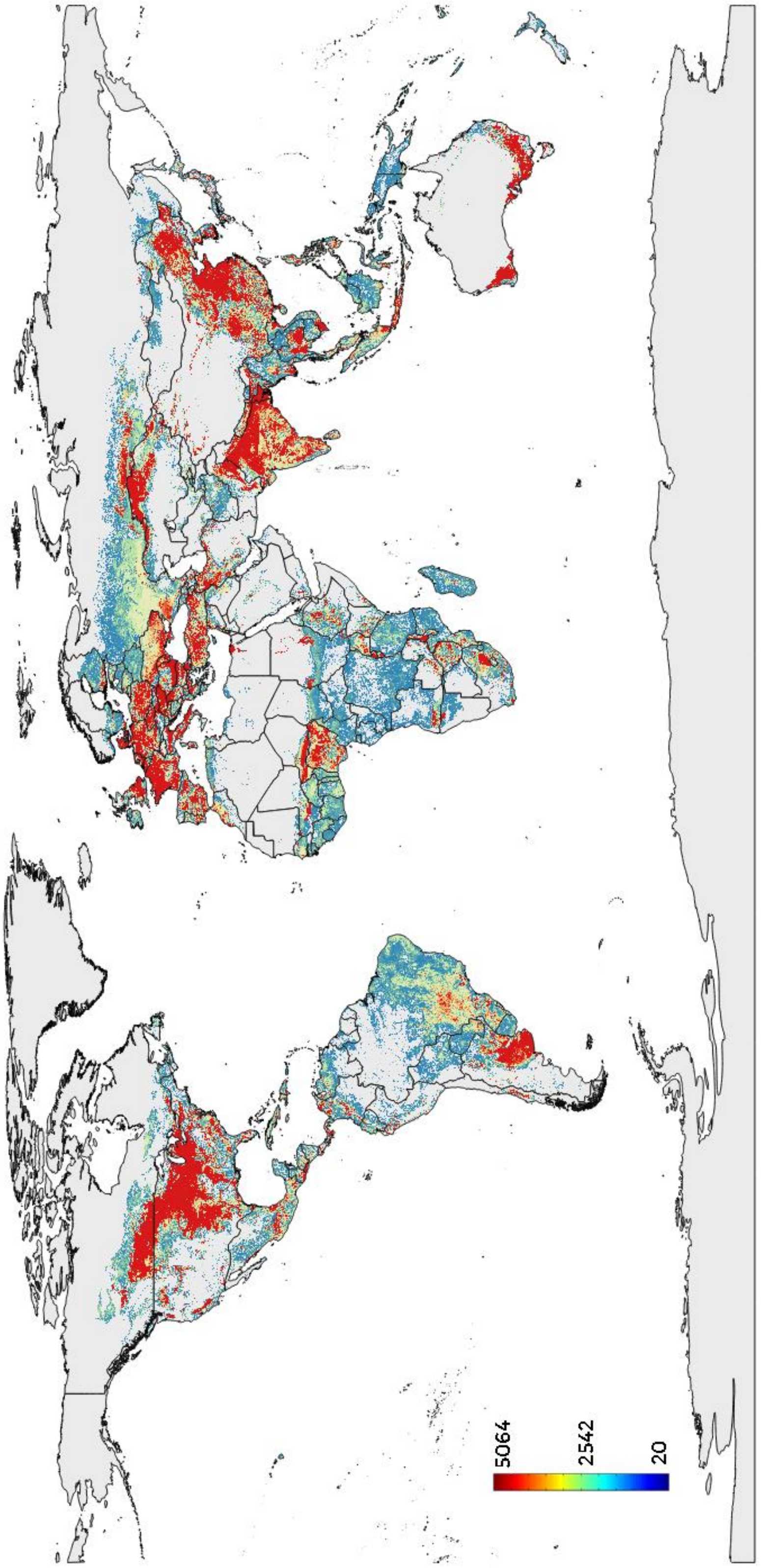
Minimize m^3_{water} utilized, maximizing gr_{prot} production

Crop distribution

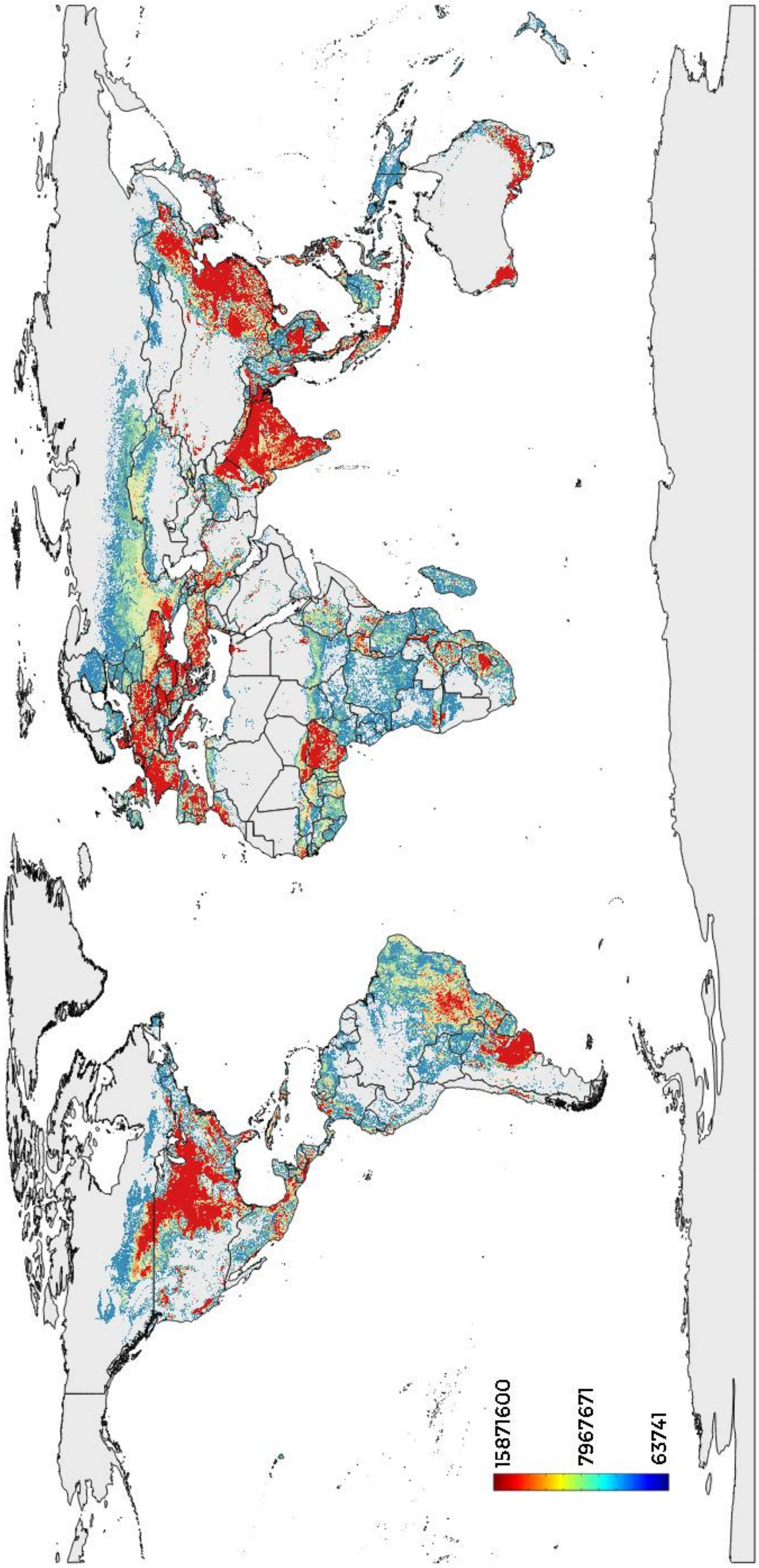


Minimize mwater3 utilized, maximizing grprot production

ton ***produced***

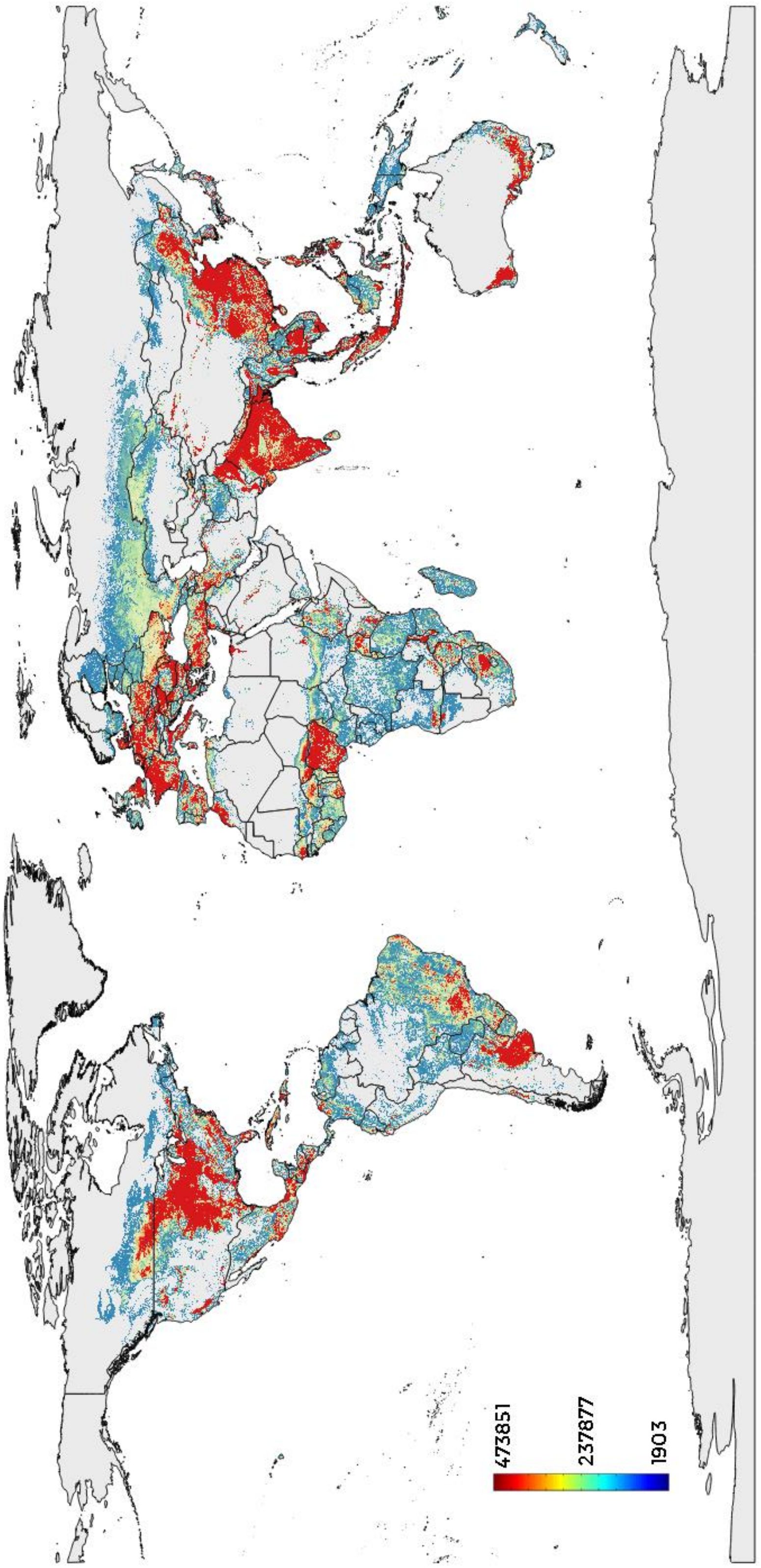


$Gcal$ ***produced***



Minimize $mwater3$ utilized, maximizing $grprot$ production

kg_{prot} ***produced***



Minimize mwater3 utilized, maximizing grprot production

m^3_{water} ***used***