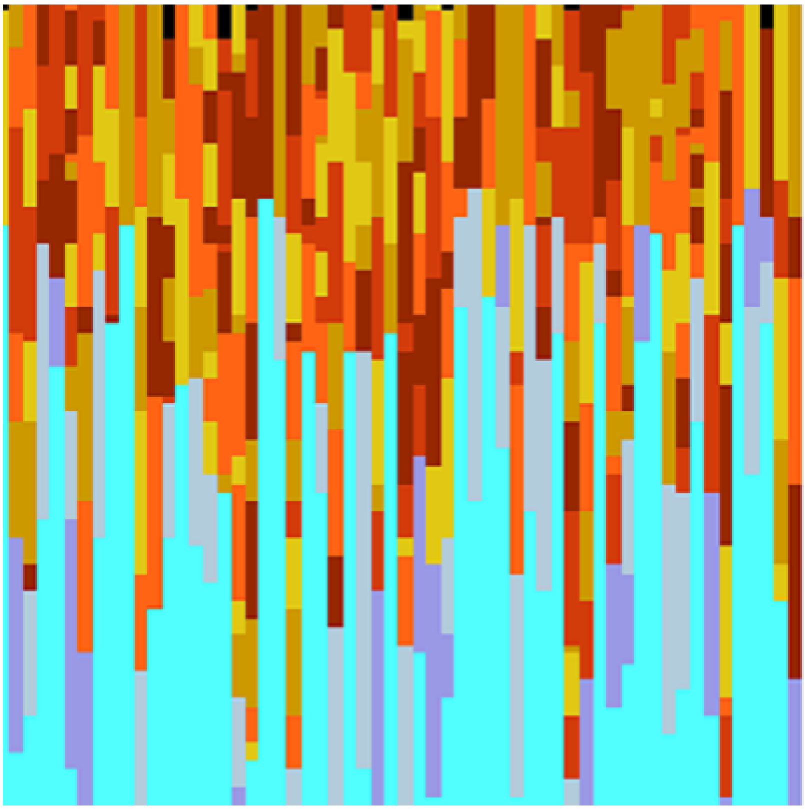
Users guide to GeoPriorApp

By Jesper Nørgaard



A tool developed for the INTEGRATE project to generate 1D subsurface realizations from user specified inputs.

Overview

1. Installation

The code for the app is written in MATLAB (2023a) a runtime installed is necessary to run the program. Below is the guide on where to get it and how to install it:

Download and install the Windows version of the MATLAB Runtime for R2023a from the following link on the MathWorks website:

https://www.mathworks.com/products/compiler/mcr/index.html

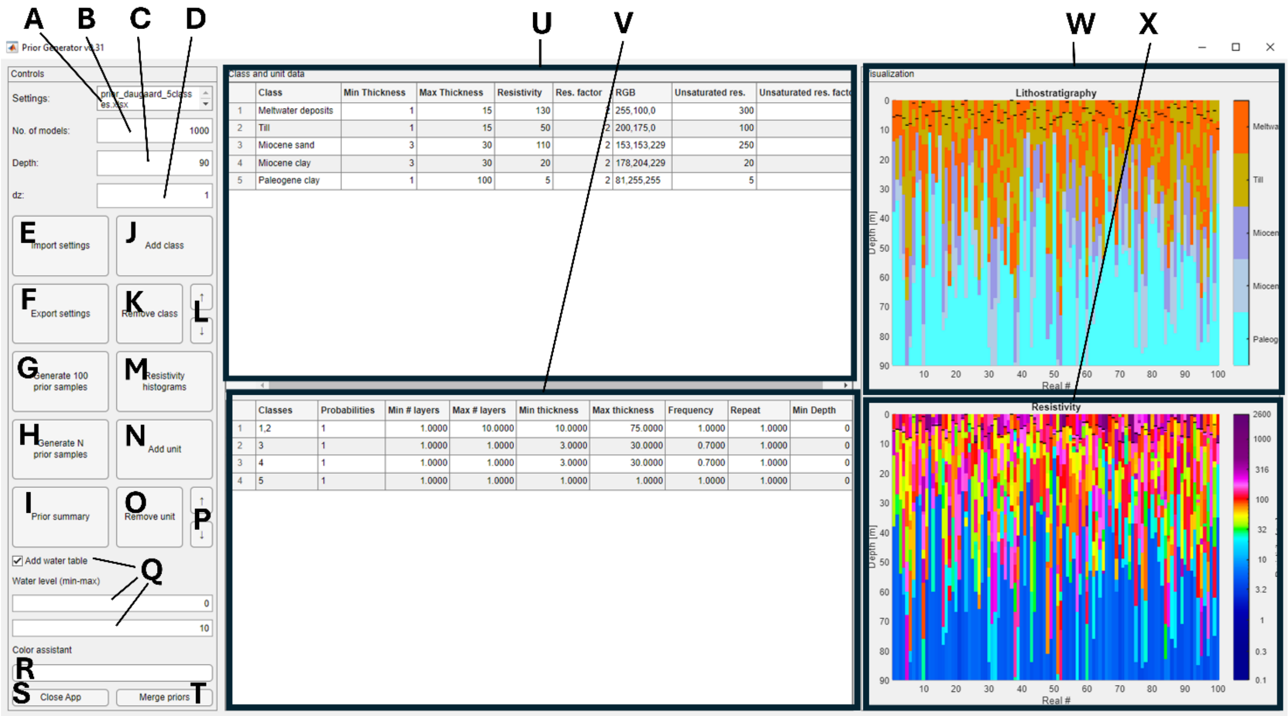
NOTE: You will need administrator rights to run the MATLAB Runtime installer.

For more information about the MATLAB Runtime and the MATLAB Runtime installer, see "Distribute Applications" in the MATLAB Compiler documentation in the MathWorks Documentation Center.

Then run the distributed “**PriorGeneratorApp.exe**”. This does not require any further installations. The program should open.

2. App Walkthrough

A button-by-button walkthrough of all interactive features in the program.



A: Filename:

Filename of the excel file that contains the prior settings. Will automatically set if settings are imported or exported. The field is mostly a reminder for the user.

B: No. of models:

Number of models to generate when generating N priors.

C: Depth:

The maximum depth of the prior models in meters. Used when generating priors.

D: dz:

The thickness of the model parameters in the prior model. Used when generating priors.

E: Open…:

Imports prior settings from an excel file. Formatting in the excel file should be exact to be read correctly. If the file was generated using the "Save as" function of the PriorGeneratorApp this will be guaranteed.

F: Save as…:

Exports the current settings in the PriorGeneratorApp window to an excel file.

G: Generate 100 prior samples:

Generates 100 prior realizations using the current settings in the PriorGeneratorApp. Also visualizes the realizations in the "Visualization" section of the app UI. The 100 generated realizations are saved to a HDF5 file labeled with the number of realizations, depth, and time of day (e.g. ‘prior_apptest_N100_dmax90_ 20250919_1449.h5’).

H: Generate N prior samples:

Generates N prior realizations using the current settings in the PriorGeneratorApp. Also visualizes the first 100 realizations in the "Visualization" section of the app UI. The generated realizations are saved to a HDF5 file labeled with the number of realizations, depth, and time of day (e.g. ‘prior_apptest_N1000000_dmax90_ 20250919_1449.h5’).

I: Prior summary:

After generating a prior file: Summarized key outcomes of the prior setup for user inspection such as mode, marginal distribution and entropy of the prior.

J: Add class:

Adds a class to the *class data table* with arbitrary properties.

K: Remove class:

Removes the selected row from the *class data table*.

L: Class up/down:

Moves the selected row from the *class data table* up or down.

M: Resistivity histograms:

Visualizes the resistivity inputs from the "Resistivity" and "Resist. Uncertainty" columns of the *class data table*. If "Add water table" is checked the unsaturated resistivities are also visualized. If not closed, the figure will live update as values are changed in the *class data table*.

N: Add unit:

Adds a unit to the *unit data table* with arbitrary properties.

O: Remove unit:

Removes the selected row from the *unit data table*.

P: Class up/down:

Moves the selected row from the *class data table* up or down.

Q: Add water table:

Adds the possibility to input properties related to the water table. Two new fields appear below that allows to specific the minimum and maximum depths to the water table in meters. Adds two additional rows to the *class data table* "Unsaturated res" and "Unsaturated res unc" that specifies the resistivity of the classes above the water table.

R: Color assistant:

The color field updates as soon as any field in the RGB column is edited. The button can be click to open up the catalogue of colors used in the Danish National well log database (Jupiter).

S: Close app:

Closes the program.

T: Merge priors:

Opens a new window where two priors generated using the app can be merged into one HDF5 file. If more than two priors wish to be merged, new priors can be merged with the already merged priors.

U: Class data table:

Input data related to the classes of the prior model.

Class name: The name to attribute this class.

Min thickness: The minimum thickness that this class is allowed to have in the prior realizations in meter (random thicknesses will be uniformly drawn between the minimum and maximum values).

Max thickness: The maximum thickness that this class is allowed to have in the prior realizations in meter (random thicknesses will be uniformly drawn between the minimum and maximum values).

Resistivity: The central value of the logarithmic-gaussian distribution of the resistivity in Ohm-m.

Resist. Uncertainty: A factor that matches three standard deviations (3-σ) in the logarithmic space. This mimic the observed distribution when pooling real resistivity measurements. The choice of 3-σ makes it easier to limit the maximum and minimum values of the distribution as 99.7% of the distribution lies within 3-σ.

RGB: The red, green, blue color code for the specific class. Values must be between 0 and 255.

Unsaturated res: The central value of the logarithmic-gaussian distribution of the class resistivity if the lithology is above the water table.

Unsaturated res unc: A factor that matches three standard deviations (3-σ) in the logarithmic space if the lithology is above the water table.

V: Unit data table:

Input data related to the subsurface architecture of the prior model.

Classes: The classes that should be generated within this unit.

Probabilities: The probabilities of the respective classes. The probabilities must sum to 1. Must have the same number of values as classes. Alternatively, the user can simply input *1* if all classes are equally likely.

Min # layers: The minimum number of layers with the unit (random numbers will be uniformly drawn between the specified minimum and maximum values).

Max # layers: The maximum number of layers with the unit (random numbers will be uniformly drawn between the specified minimum and maximum values).

Min thickness: The minimum thickness of the unit (random thicknesses will be uniformly drawn between the minimum and maximum values). Keep in mind that there must be some consistency between the thicknesses chosen in the *class data table* and the thickness of the units. If not consistent the model might warp the uniform distribution of the class thicknesses severely.

Max thickness: The maximum thickness of the unit (random thicknesses will be uniformly drawn between the minimum and maximum values). Keep in mind that there must be some consistency between the thicknesses chosen in the *class data table* and the thicknesses of *the unit data table*. If not consistent the model might warp the uniform distribution of the class thicknesses severely.

Frequency: How often the unit should be generated in the prior model: 0 means the unit is in 0% of the generated models, 1 means that the unit is in 100% of the generated models.

Repeat: Whether the same class is allowed to be drawn two times in a row. If 1 the next layer is completely randomly drawn. If 0 the next layer drawn must always be different than the previous layer.

Min Depth: The minimum depth a unit is allowed to be present. Caution is advised when using this input as miss-matches with other inputs can cause a lot of redraws.

W: Prior class visualization:

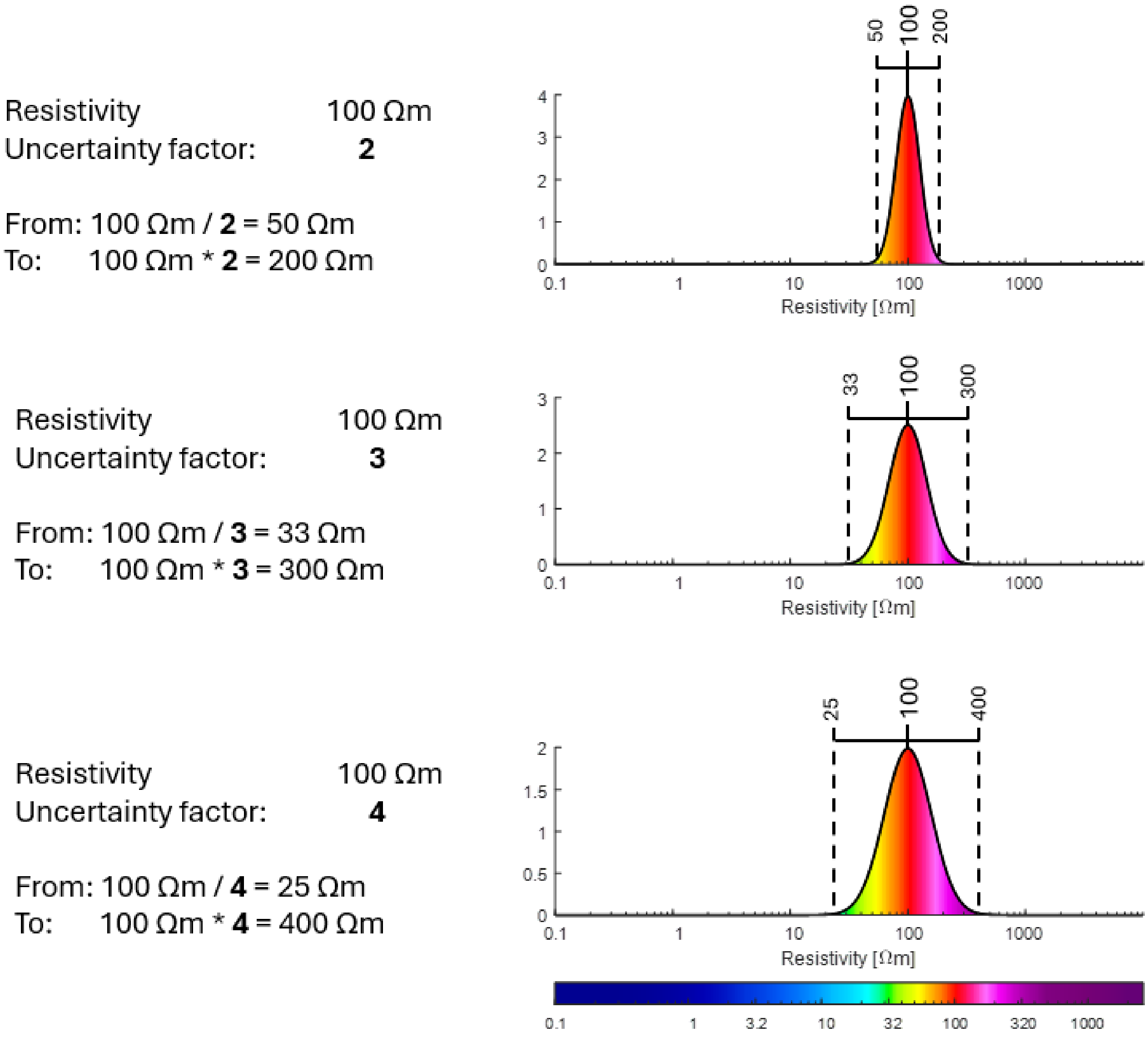
Visualizes the lithologies of the first 100 prior models generated. If water table is part of the prior model it will be represented with small black lines indicating the depth to the water table. The x-axis is the prior realization numbers, and the y-axis is the depth.

X: Prior resistivity visualization:

Visualizes the resistivity of the first 100 prior models generated. If water table is part of the prior model it will be represented with small black lines indicating the depth to the water table. The x-axis is the prior realization numbers, and the y-axis is the depth. The colorscale is specially designed for resistivity visualization.

3. Note on the resistivity uncertainty formulation

Below is a figure that can be used as guidance when considering the resistivity uncertainty:



4. Output HDF5 file structure

Dataset	Attribute	Description	Structure	Format
M1		Matrix holding resistivity values of N realizations with Nm model parameters	[N,Nm]	Single
M1	is_discrete	Described whether /M1 is a discrete (**1**) or continuous parameter (**0**)	0	Double
M1	x	Array of z values (top of layer) for M1	[Nm]	Double
M1	clim	Min and max values of the colorbar	[1,2]	Double
M1	cmap	Colormap with nlev levels	[3,nlev]	Double
M1	name	What is M1	'Resistivity'	String
M2		Matrix holding lithology values of N realizations with Nm model parameters	[N,Nm]	Int16
M2	is_discrete	described whether /M2 is a discrete (**1**) or continuous parameter (**0**)	1	Double
M2	class_name	Names of all lithology classes	[n_class]	Double
M2	class_id	Identifiers for all lithology classes	[n_class]	Double
M2	x	Array of z values (top of layer)	[Nm]	Double
M2	clim	Min and max values of the colorbar	[1,2]	Double
M2	cmap	Colormap with nlev levels	[3,n_class]	Double
M2	name	What is M2	'Lithology'	String
M3[a]		Matrix holding values for the water table depth (in meters)	[Nm]	Single
M3[a]	is_discrete	described whether /M3 is a discrete (**1**) or continuous parameter (**0**)	0	Double
M3[a]	Name	What is M3	'Waterlevel'	String
M4[b]		Vector holding the prior adherence of the realizations	[Nm]	Single
M4[b]	is_discrete	described whether /M4 is a discrete (**1**) or continuous parameter (**0**)	0	Double
M4[b]	class_name	Names of the original prior files merged	[n_class]	Double
M4[b]	class_id	Identifiers for merged priors	[n_class]	Double
M4[b]	x	z values (mandatory)	0	Double
M4[b]	clim	Min and max values of the colorbar	[1,2]	Double

M4[b]	cmap	Colormap with n_priors levels	[3,n_priors]	Double
M4[b]	name	What is M4	‘Prior’	String

[a] M3 is only written to the HDF5 file if the “Add water table” box is checked.

[b] M4 is only written to the HDF5 file if it is merged from two other files.

5. Settings excel file structure

The structure of the excel file is strict. Tab names and headlines needs to be specific, otherwise the program cannot load the settings. Below is an overview of the headlines that are under the tabs in the excel sheet. Note that neither the number of simulations, the depth, or the step size are part of the prior settings stored in the excel file. For more details on the function of each headline refer to section 2.

Geology1

Class	Max thickness	Min thickness	RGB color

Geology2

Classes	Probabilities	Max no of layers	Max no of layers	Min unit thickness	Max unit thickness	Frequency	Repeat	Min depth

Resistivity

Class	Resistivity	Resistivity uncertainty	Unsaturated resistivity	Unsaturated resistivity uncertainty

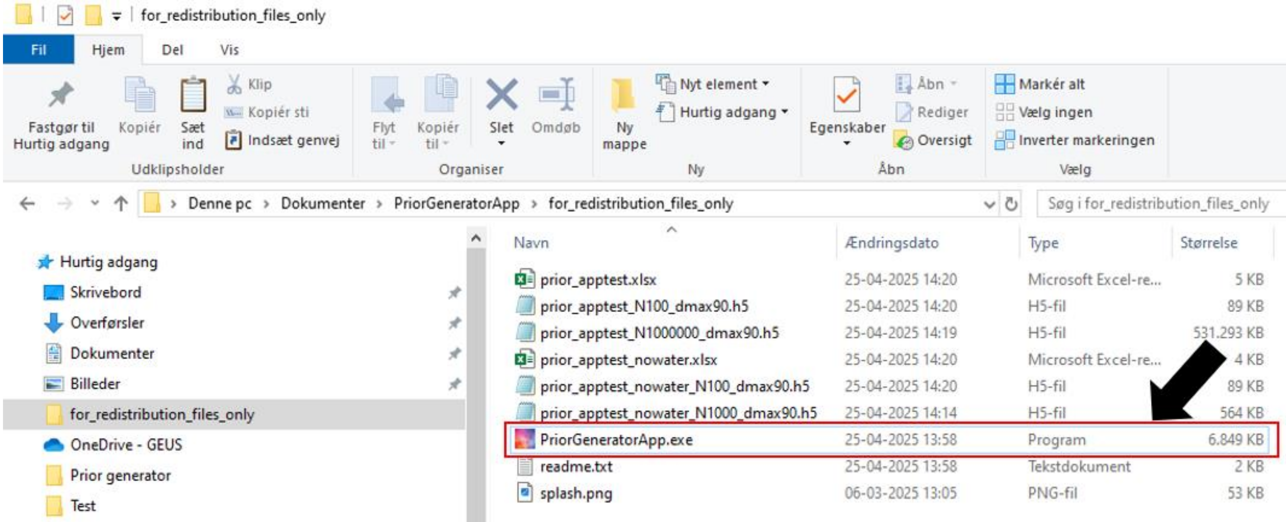
Water table

Min depth to water table	Max depth to water table

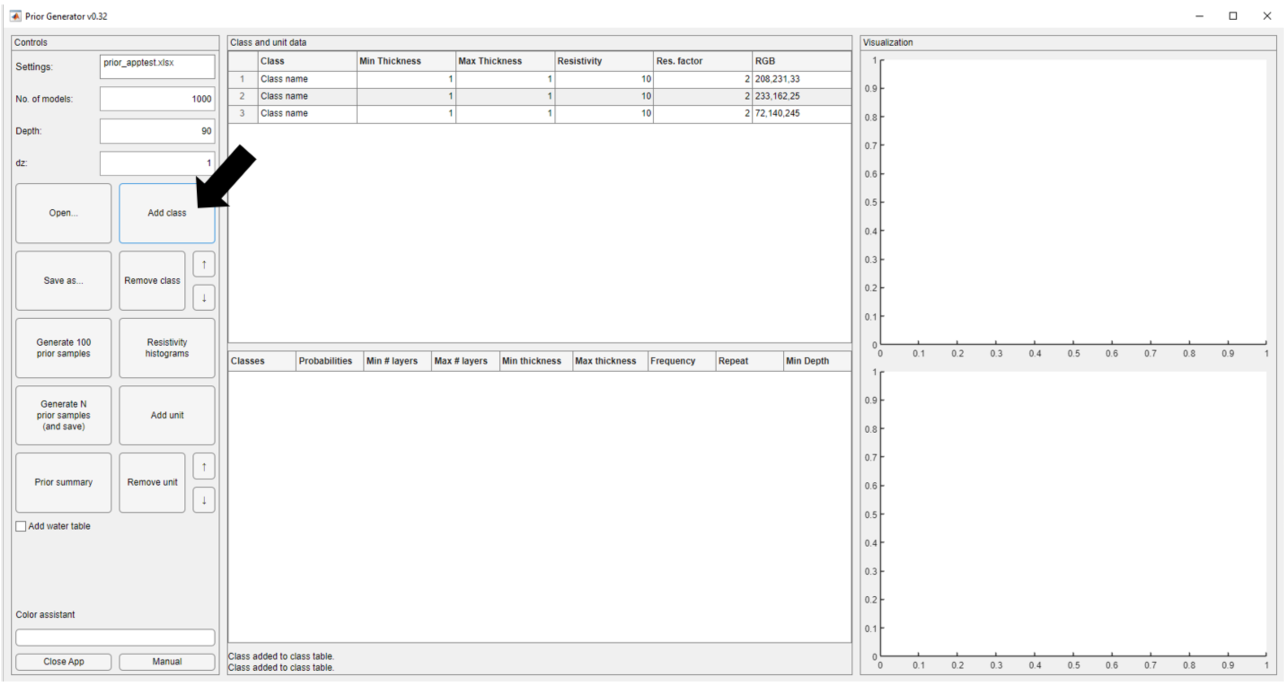
6. Example

Next is a guided example you can follow to get to know the different functionalities of the program.

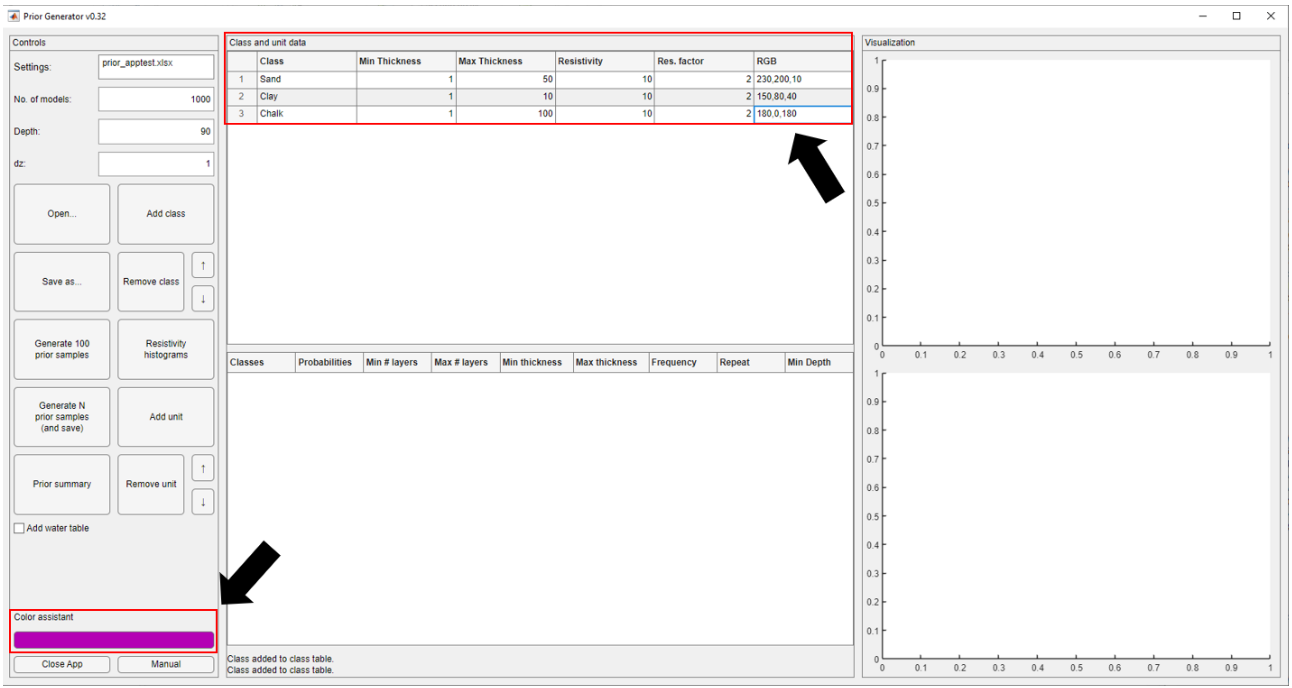
1. Locate and run the executable file.



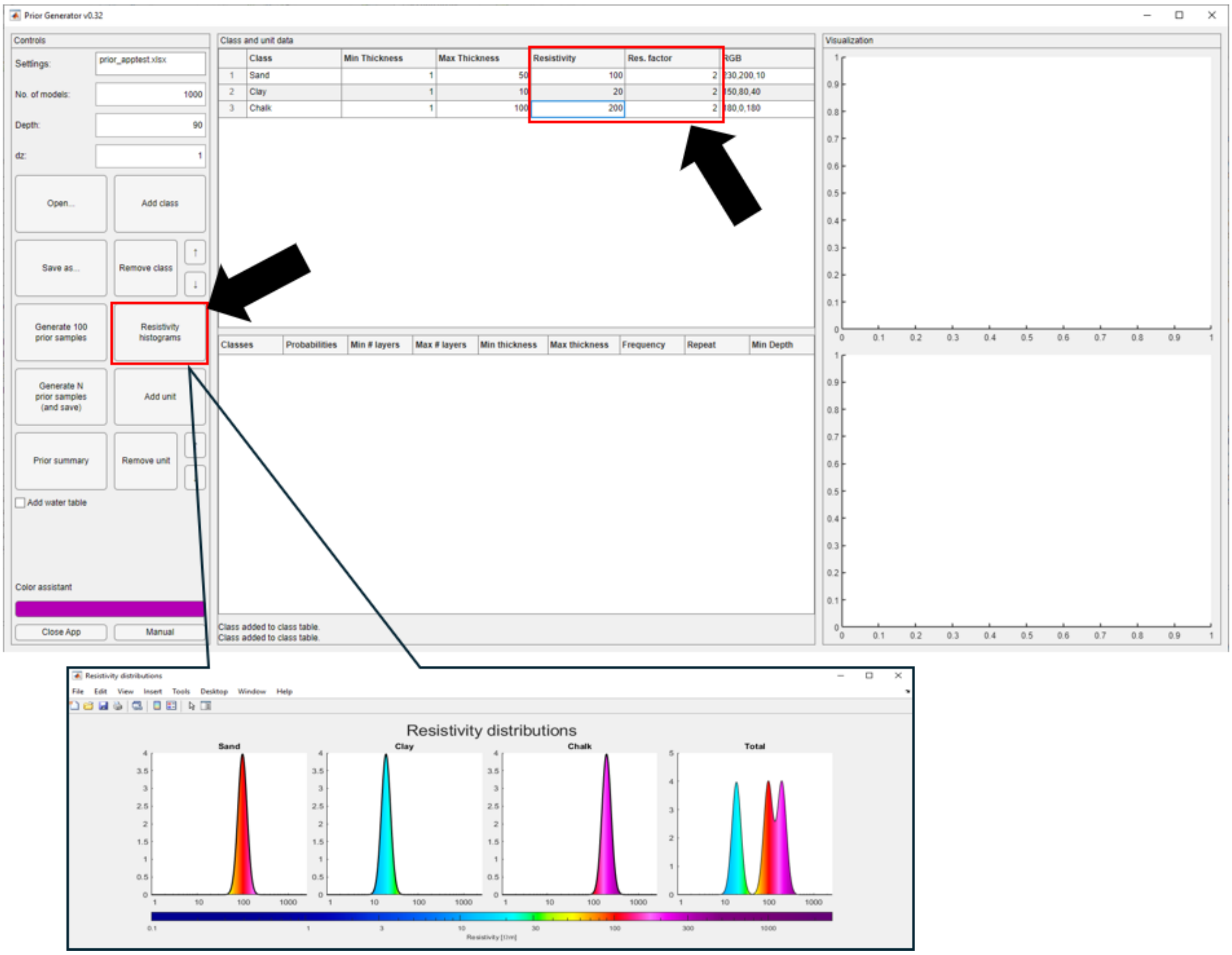
2. This walkthrough will work with three classes. Press the 'Add class' button three times.



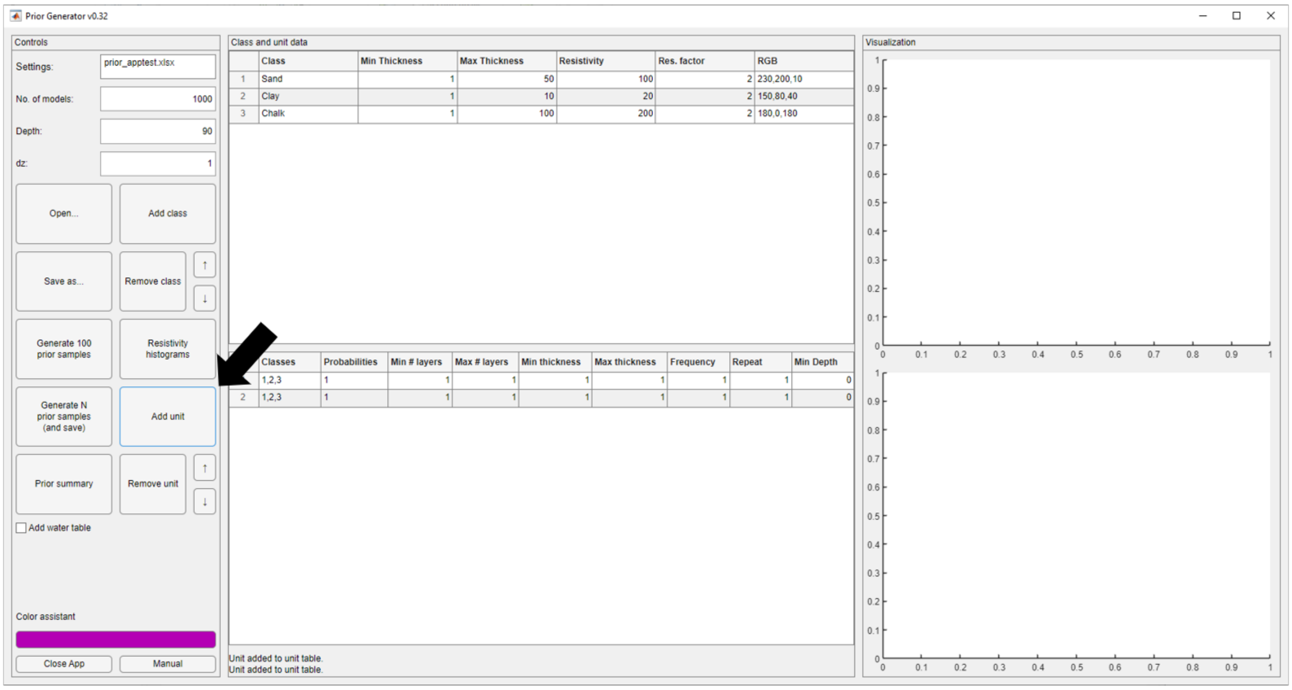
3. Three rows are now created in the class data table. Now we need to edit them for our purpose. Our example considers sand, clay, and chalk layers. We know that our sand layers can be relatively thick (1-50 m), our clay layers are a bit thinner (1-10 m), and the chalk can be very thick and should be at the bottom of our models. We also want to adjust the color of our layers so that sand is yellow, clay is brown, and chalk is purple. The color assistant will help us see the colors we've chosen.



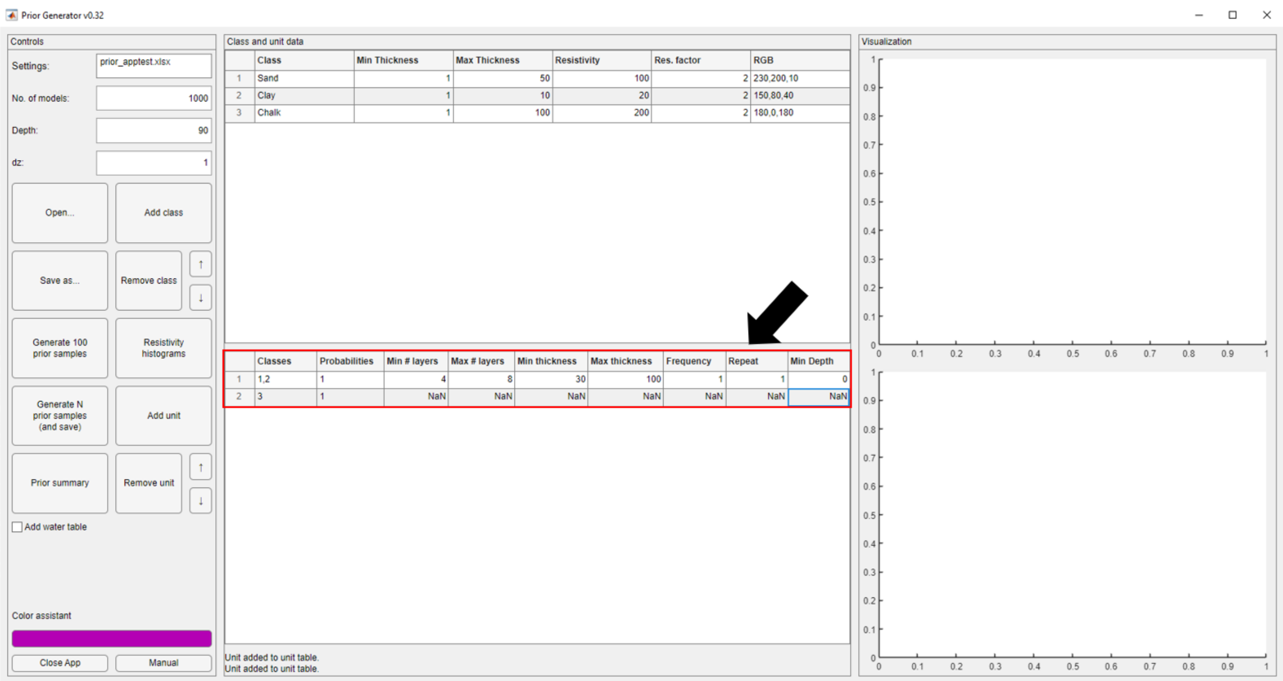
4. We also need to assign resistivities to the three classes. Her we will chosen 100 Ωm for the sand, 20 Ωm for the clay and 200 Ωm for the chalk. To start we set the resistivity uncertainty of all three classes to 0.05. To inspect the resistivity values we have chosen we press the 'Resistivity histograms' button.



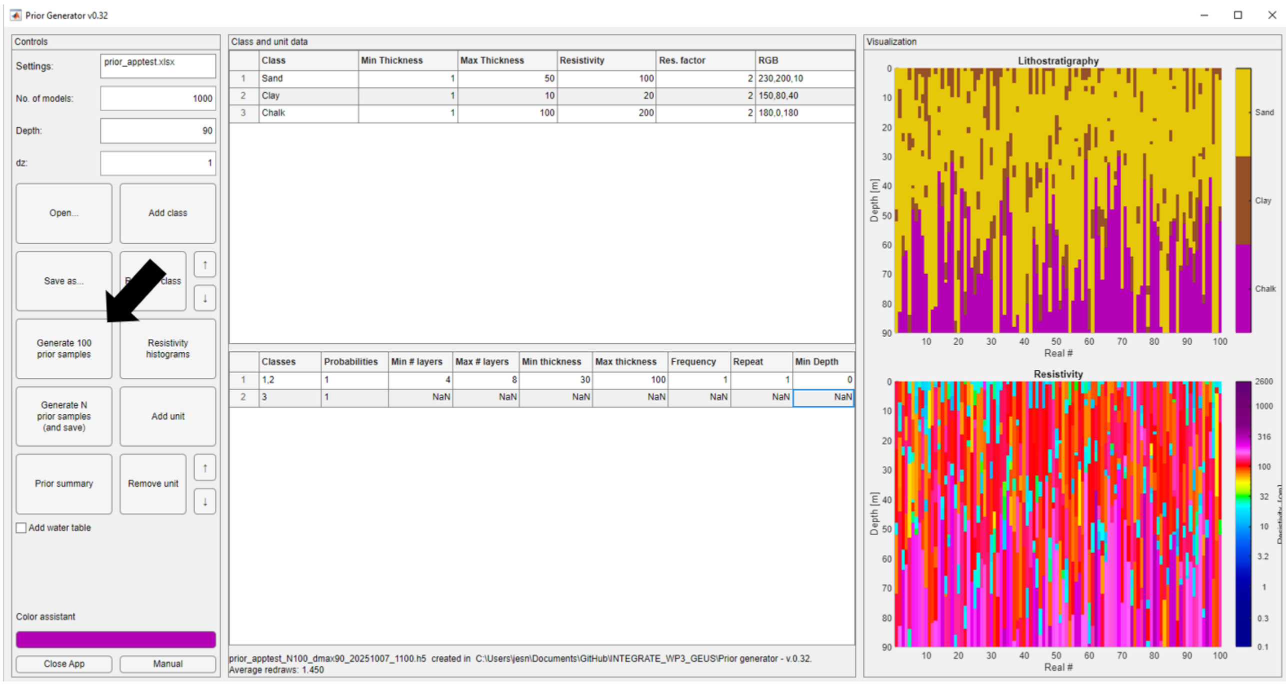
5. Now we need to use the unit data table to arrange the chosen classes in a stratigraphic order. We will add two units, 1) contain sand and clay layers, this could for example be deposits on a fluvial plain. 2) contains only the chalk layer, this could be the underlying bedrock in the area. We press 'Add unit' two times.



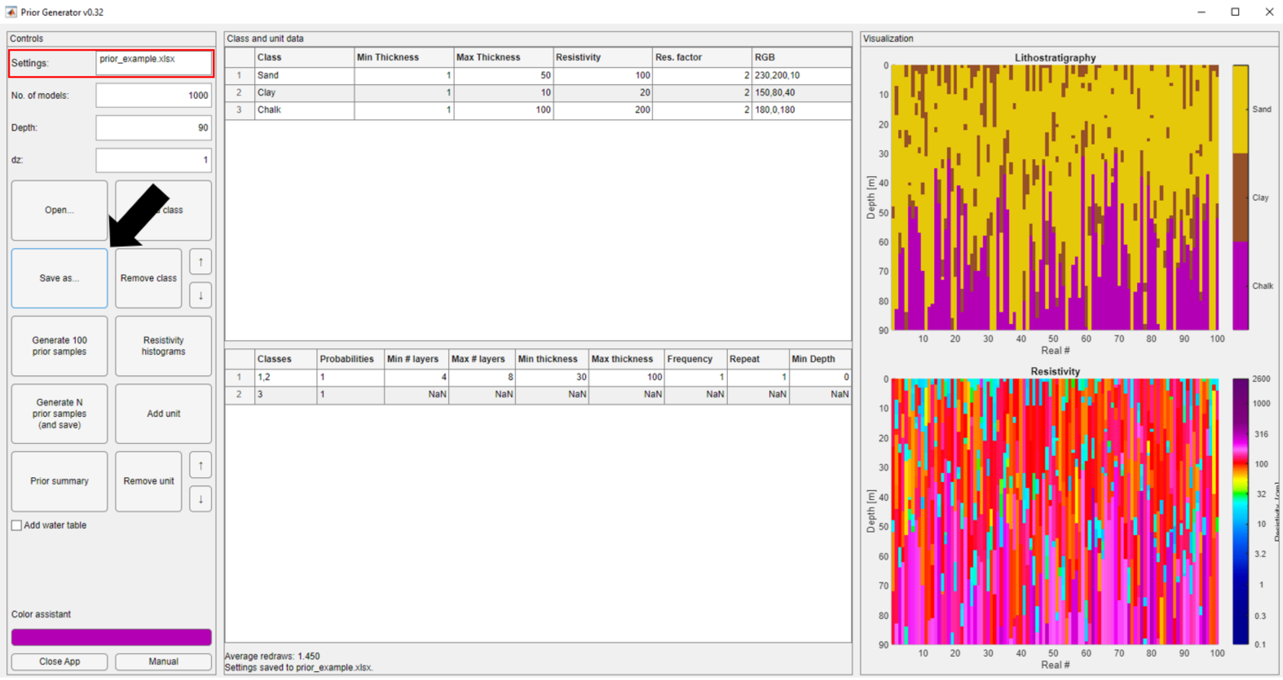
7. We enter the classes 1 and 2 (sand and clay) in the first unit. We let the models minimum have 4 layers in this sequence and maximum 8. We want the unit package to be at least 30 meters thick and allow it to become 100 meters thick. In the second unit we enter just 3 (chalk) as this is the underlying geology. When you are working with the last unit in the sequences the only relevant property is the classes appearing in it, as it will always fill our the remainder of the sequence.



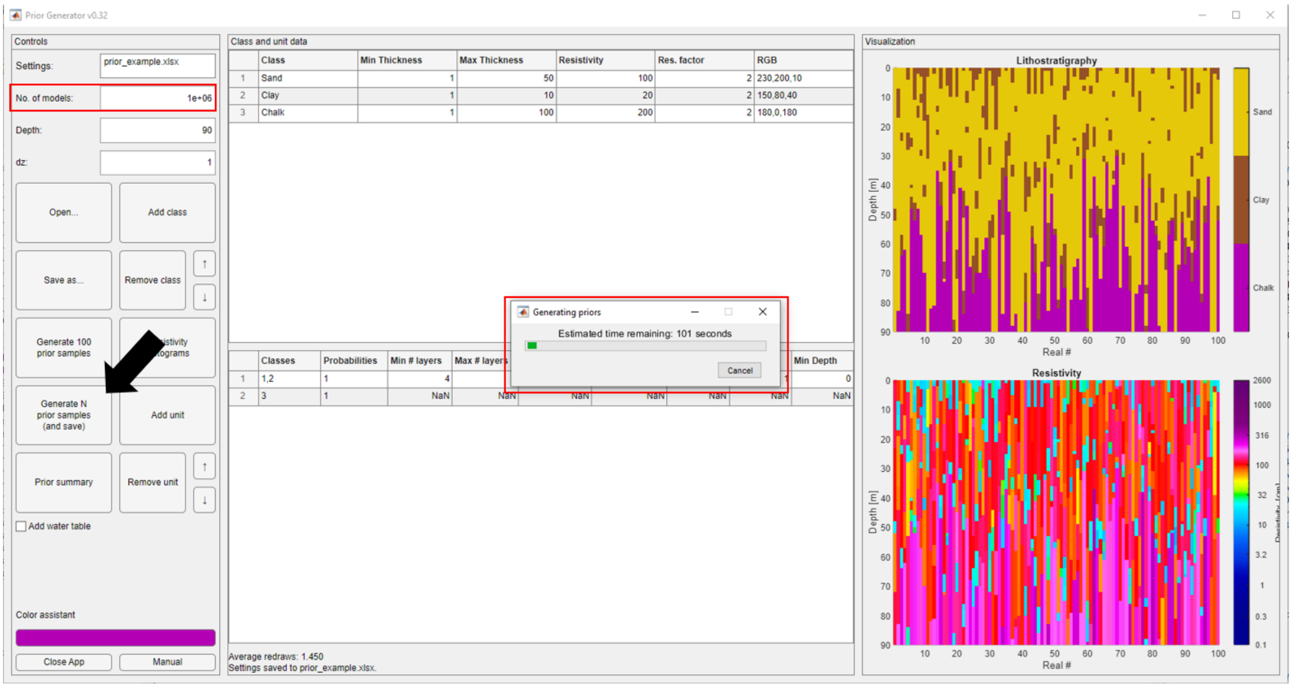
8. We have now input enough information to generate an ensemble of prior realizations. To see how we are doing we press the 'Generate 100 prior samples' button.



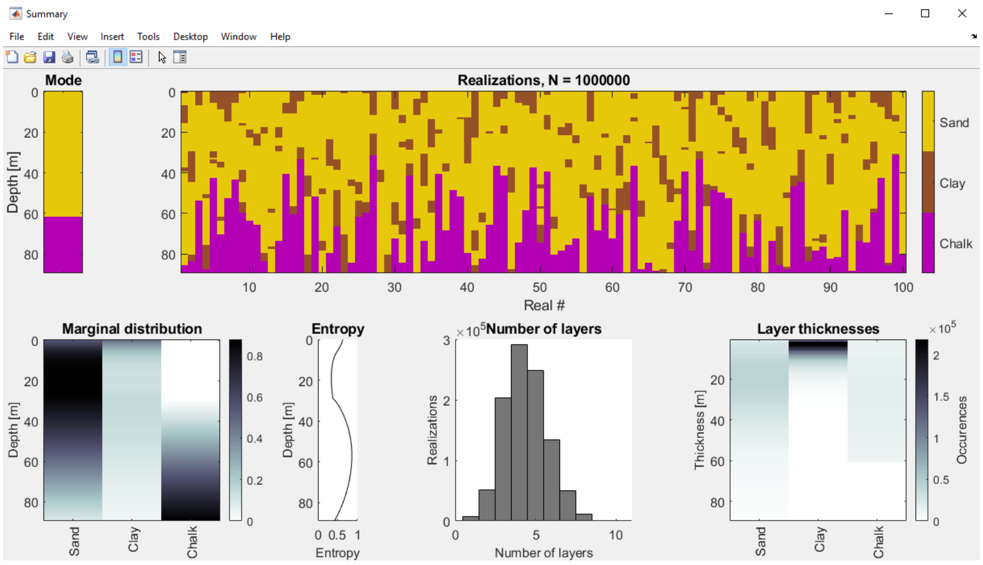
9. We are happy we the settings we've chosen and therefore want to save them for later. We press the 'Save as…' button and save the file as 'prior_example'. You will see the working filename change. If we at a later point want to edit this prior setup, we simple open the excel file using 'Open..' button.



10. Since we've decided that we are happy with the setup we want to generate more realizations. To do this enter the desired numbers of realizations in the 'No. of models' field. Then press the 'Generate N prior samples' button. A progress dialog will open to show the time remaining. Once finished a pop-up will show you have successful the operation was.



11. Lastly, we want to inspect the prior generated in more detail. To do this we can utilize the 'Prior summary' button. This will open a new window with advanced statistics we can study about the prior.



12. Congratulations you have now generated your first prior file *prior_example_N1000000_dmax90_XXXXXX_XXXX.h5*. Find it in your workfolder and use the brand-new look-up table in your next probabilistic inversion!

6. MATLAB

Guide to generating hdf5 prior files:

1. Fill in excel sheet with desired values and according to section 4. Alternatively generate the excel file using the Prior Generator App.
2. Run prior_generator.m
3. Find hdf5 file located in the folder

prior_generator.m:

Generates a hdf5 file from the prior model excel spreadsheet.

Called using `prior_generator(input,Nreals,dmax,dz,doPlot)`

`input`	Name of excel spreadsheet that defines prior (needs to be in same folder)
`Nreals`	Number of realizations to output
`dmax`	Depth of the models in meter
`dz`	Layer thicknesses of model parameters in meter
`doPlot`	1 will display 100 realizations, 0 no plot

Other notable MATLAB files found in the repository:

get_prior_sample.m:

Is called by prior_generator.m and produces N realizations from the given inputs.

prior_lith_reals.m:

Is called by get_prior_sample.m and assigns lithologies to the realizations.

prior_res_reals.m:

Is called by get_prior_sample.m and assigns resistivities to the realizations.

prior_water_reals.m:

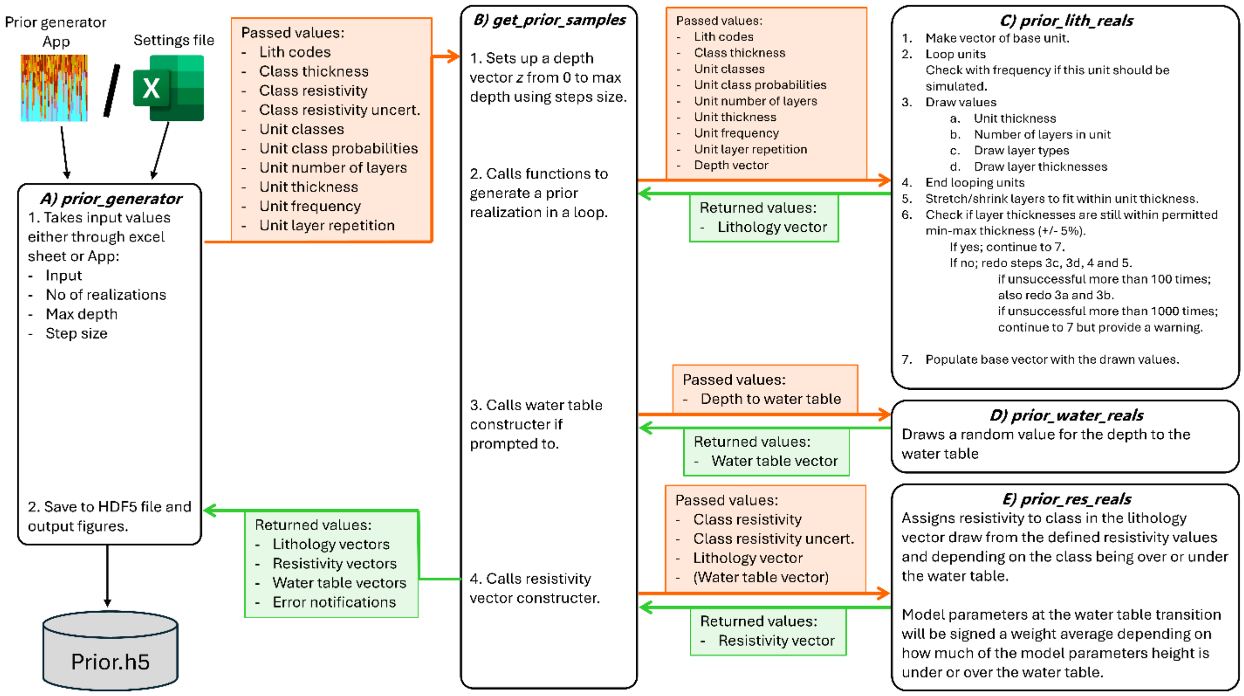
Is called by get_prior_sample.m and assigns a water table to the realizations.

flj_log.m:

Contains the resistivity colormap.

count_category_all.m, distribution_stats.m, and entropy.m:

Are functions used to generate prior summary statistics.



Flow diagram of the primary functions of prior generator files.

7. Python

Guide to generating hdf5 prior files:

1. Fill in excel sheet with desired values and according to section 4. Alternatively generate the excel file using the Prior Generator App.
2. Run prior_generator.py
3. Find hdf5 file located in the folder

prior_generator.py:

Generates a hdf5 file from the prior model excel spreadsheet.

Called using `prior_generator(input,Nreals,dmax,dz,doPlot)`

`input`	Name of excel spreadsheet that defines prior (needs to be in same folder)
`Nreals`	Number of realizations to output
`dmax`	Depth of the models in meter
`dz`	Layer thicknesses of model parameters in meter
`doPlot`	1 will display 100 realizations, 0 no plot

Other notable Python helper functions in the repository:

get_prior_sample.py:

Is called by prior_generator.py and produces N realizations from the given inputs.

prior_lith_reals.py:

Is called by get_prior_sample.py and assigns lithologies to the realizations.

prior_res_reals.py:

Is called by get_prior_sample.py and assigns resistivities to the realizations.

prior_water_reals.py:

Is called by get_prior_sample.py and assigns a water table to the realizations.

flj_log.py:

Contains the resistivity colormap.