

User's Guide for CamOptimus Graphical User Interface

CamOptimus is a tool for applying Genetic Algorithm (GA) to solve multi-parametric optimisation problems and Symbolic Regression (SR) to obtain models using the data generated during optimisation procedure to investigate the effect of individual parameters on the system of interest. The source code for the compiled software and the Graphical User Interface (GUI) of the application are available under free licensing (GNU General Public License v3.0) at (<https://doi.org/10.17863/CAM.10257>). The MATLAB Runtime environment (version 9.0.1) allows the executable versions (the compiled version and the GUI) to work on computers running on Windows OS or Mac OSX. The MATLAB Runtime is available under free licensing at <http://uk.mathworks.com/products/compiler/mcr/>.

The user first selects the action to be taken in the main page of the GUI; using Genetic Algorithm to solve a multi-parametric optimisation problem or conducting Symbolic Regression analysis to investigate a given solution sub-space. Each action is comprised of successive steps and the decisions made in each step feeds into the next one, guiding the user through the course of action to be taken. This documentation provides a detailed explanation of each step as well as the format regarding the inputs and outputs.

The main interactive window of the tool is:



A. GENETIC ALGORITHM

Once the Genetic Algorithm option is selected in the main page, the following interactive dialogue appears:


The screenshot shows the 'CamOptimus: Genetic Algorithm' window. The interface is divided into several sections:

- Initial Setup:**
 - Questions: 'Are you running your first generation of experiments?' (Yes), 'If you have more than one objective, are they equally important?' (Yes), 'Would you like to include factors that will not be optimized but will appear in our generated reports?' (Yes), and 'Are you planning to change this factor?' (Yes).
 - Objective: A table with columns 'Name' and 'Weight'. The first row has 'Objective' and '1'. A 'Set objective' dropdown is set to 'Maximize'.
 - Buttons: 'Add or change objective' and 'Remove objective'.
- Factors list:** A table with columns 'Factors Name', 'Value', and 'Units'. Rows include 'Absolute', 'Minimum', and 'Maximum'. Buttons: 'Add or change factor' and 'Remove factor'.
- Advanced settings:**
 - Question: 'Would you like to change the default genetic algorithm parameters?' (Yes).
 - Parameters: 'Mutation rate', 'Crossover rate', 'Number of bits', and 'Selection probability'.
 - Buttons: 'Default' and 'Accept'.
- Buttons:** 'Load parameters', 'Load Experiments', 'Plot results', and 'Save results'.
- Population profiling of the factors:** A plot with 'Frequency' on the y-axis (0 to 1) and an x-axis from 0 to 1.
- Best performing fraction:** A plot with 'Frequency' on the y-axis (0 to 1) and an x-axis from 0 to 1.
- Would you like to generate new experiments?** (Yes) with 'Generate experiments' and 'Save new Experiments' buttons.
- Bottom buttons:** 'Generate experiments', 'Save parameters', and 'Save Experiments'.

This interactive interface allows actions to be taken in a given order so as to guide the user through the steps of the procedure. It is comprised of two blocks of information; one for setting up the experiment and another for evaluating the results.

1. Describing the initial setup

There is a set of information required to be determined and fixed constant throughout the application and therefore, the first question that needs to be answered is to select whether the first generation of experiments will be carried out or not. Selection questions open up with a

drop-down menu and the selection is accepted by the Enter (). The information the experimenter is asked to provide are regarding the objective(s) of the experiment and the factors of interest, which are thought to have an impact on the objective(s). There are several GA-associated parameters, which the user might wish to alter, and the interface allows these changes under the “Advanced settings”.

a. Defining the objective(s)

The first decision to be made in an optimisation experiment is to define the measurable objective(s) that are wished to be optimised. If there is more than one objective, the first question that the user needs to address is whether these objectives are equally important, and hence could be assigned equal weights, for the setup under investigation or not. If the answer to the question is “Yes”, the user only has to provide a name for each objective and select whether that target objective should be maximised or minimised. Combinations of objectives where some are minimised where others are maximised are allowed. Once the “Add or change objective” button is hit, the individual objective should appear in the Objectives list to the right of the setup. An objective already defined to the system can be removed by first selecting that objective in the list and then hitting the “Remove objective” button. If individual weights are to be assigned by the user, the “Weight” box will also appear highlighted, so that it can be modified. One important thing to note is that if an objective is removed from a setup where the user defines individual weights, all weights are automatically reset to 1 (i.e. equal weights) and the user needs to redefine relative weights for the remaining objectives.

A sample entry is shown in the next page:

Initial Setup

Are you running your first generation of experiments? Yes

If you have more than one objective, are they equally important? No

Objective	Name	Weight
	c	0.2

Set objective Minimize

Add or change objective Remove objective

Objectives list

- a
- b
- c**

b. Defining factors

As soon as any objective is defined, modifications are then allowed in the next section. This section asks the user to identify all factors that need to be optimised to achieve the described combined objective.

The experiments to be conducted will be generated as a report of this initial setup. The user may wish to include additional parameters, which will not be optimised in the procedure but will be kept at a fixed value, just to have them appear on the generated reports for convenience. Such an example would be that the user may wish to optimise only a given fraction of the medium components whereas, in the lab, all components need to be included in the actual experiments. Including these unchanged parameters in the factors list will allow a full medium recipe to be generated automatically to facilitate lab work. The first question the user is asked in this section addresses this issue. The interface will allow additional factors to be included if the answer “Yes” is selected. Then for each individual factor the user has to select whether this new factor will be a factor to be optimised (the following question answered as “Yes”) or will just be monitored (the following question answered as “No”). For factors, which will be optimised, a range should be defined to vary the values in. For those that will be monitored, a single value and a corresponding unit should be introduced. The units for the inputs to the maximum and the minimum should be consistent. The software does not accept most symbols and operators such as “/”, which is frequently employed in units, so we recommend using “_” instead and omitting their use whenever possible. The

units entered by the user will automatically accommodate “_” to replace “/” or “-1”. Addition/modification/removal of factors is carried out similar to that of the objectives.

A sample entry is shown below:

The interface consists of the following elements:

- Two dropdown menus at the top:
 - Question: "Would you like to include factors that will not be optimized but will appear in our generated reports?" (Selected: Yes)
 - Question: "Are you planning to change this factor?" (Selected: Yes)
- A table for factor details:

	Value	Units
Factors Name	bbbb	
Absolute		
Minimum	26	g_ml
Maximum	89.3	g_ml
- Buttons at the bottom: "Add or change factor" and "Remove factor".
- A "Factors list" on the right showing a scrollable list: a, eeee, dddd, **bbbb** (highlighted), cccc.

c. Advanced settings

Genetic Algorithm is a search heuristic that can be successfully applied to many problems. Therefore the parameters intrinsic to its nature might need adjustment accordingly, based on similar type of problems to which it was successfully applied before. Although we believe that the parameters provided by default here are suitable for many experimental design problems encountered in biology, they should nevertheless be approached by caution in each optimisation problem. Therefore the user is allowed to make a selection either to accept the default parameters (by selecting “Yes”) or to provide values for the mutation rate (between 0 – 1), the crossover rate (between 0 – 1), the number of bits (any integer number) and the selection probability (between 0 – 1). When the default parameters are accepted, the default values will appear and the interface will not allow the user to make any further changes. If however, the user assigns values to these parameters, the values need to be stored by clicking the “Accept” button. The default settings appear as follows on the interface:

Advanced settings

Would you like to change the default genetic algorithm parameters? No ▼

Mutation rate	Crossover rate	Number of bits	Selection probability	
<input type="text" value="0.01"/>	<input type="text" value="0.9"/>	<input type="text" value="5"/>	<input type="text" value="0.5"/>	<input type="button" value="Default"/> <input type="button" value="Accept"/>

2. Generating the initial set of experiments

At this point, the user is ready to generate the first set of experiments in the optimisation procedure. Once the experiments are generated, a dialogue box appears:



These steps outlined in sections 1.a, 1.b and 1.c constitute the foundation stones for the study. Once they are designed and the experimental procedure has been initiated, they cannot be changed or modified in any way. The interface also ensures this by prompting to save this complete setup to a user-defined location in the computer (.mat extension), and to be uploaded into the system each time the tool is used for the same experiment.

The file (test_setup.mat) is:



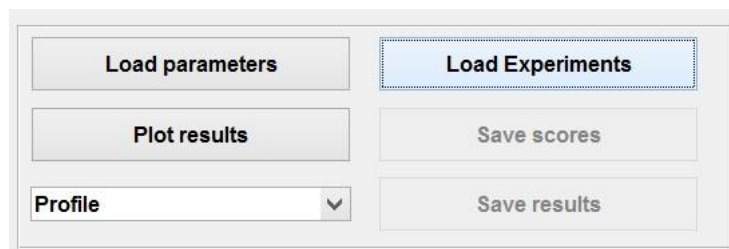
The user is then ready to save the first generation of experiments to a location of their choice in the computer (.xls extension). For a simple system of 2 objectives (a to be maximised and b to be minimised, equal weights of 1) and 3 factors; 1 kept constant (aa (units in g/L)) and two varied, bb (units in sec), and cc (unitless)) the file looks as follows:

	A	B	C	D	E	F
1	Factors	aa_g_L	bb_sec	cc_	a_1_max	b_1_min
2	Experiment 1.1	5	95.49203384	6.396517224	0	0
3	Experiment 1.2	5	47.74979925	6.735972992	0	0
4						

This file contains the first generation of experiments (all numerically coded as Experiment 1.x designating the first generation) that needs to be carried out describing the condition for each experiment in a single row (each designated by .x). It is designed in such a way that the user can take the printout to the lab to carry the experiments out. The measurable outputs for the objectives under investigation (initially generated as zeroes, later to be modified by the user) will also be recorded in the same worksheet. In case the experiments are carried out in replicates, the mean or the median values (as seen fit by the user) should be recorded in the spreadsheet. Once the experiments are carried out and the objective outcomes are recorded in the worksheet, the user is then ready to use the tool for evaluating their results and generating the next set of experiments if they see necessary.

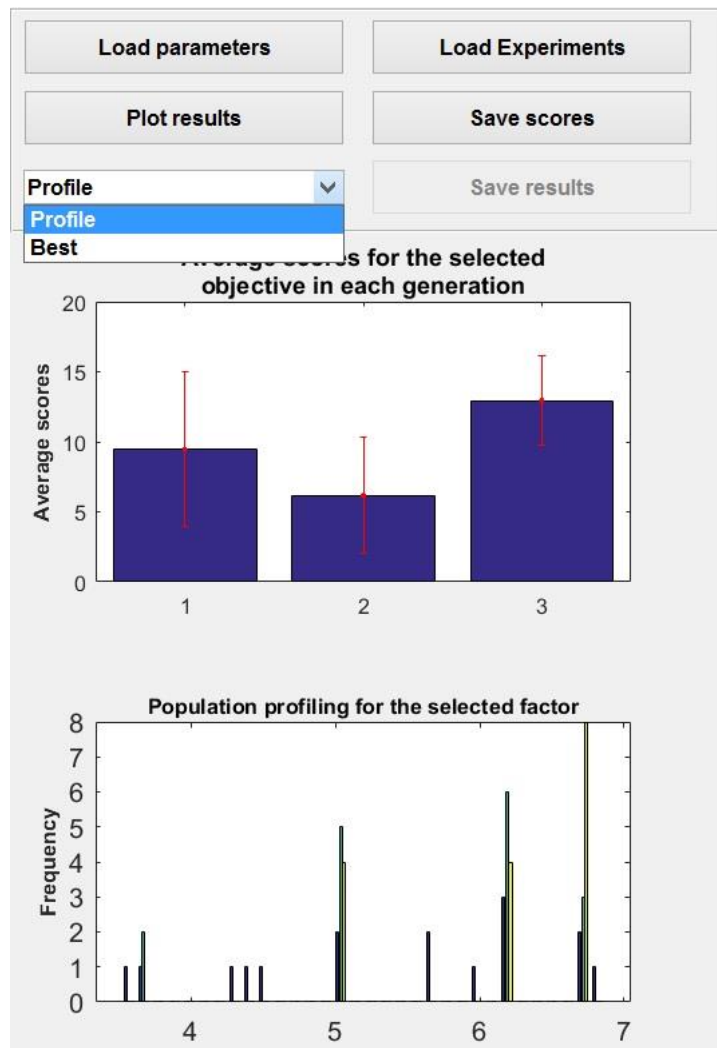
3. Evaluation of the results

Accessing the software for the upcoming generations requires uploading the two files generated in the initial setup to access the experiment under investigation. This time the user should answer the question: “Are you running your first generation of experiments?” as “No”. “Load parameter” button on the right column is activated and the parameter file with the .mat extension should be selected in the Directory as prompted. “Load experiments” button will then become active and the data file with .xls extension should then be loaded.

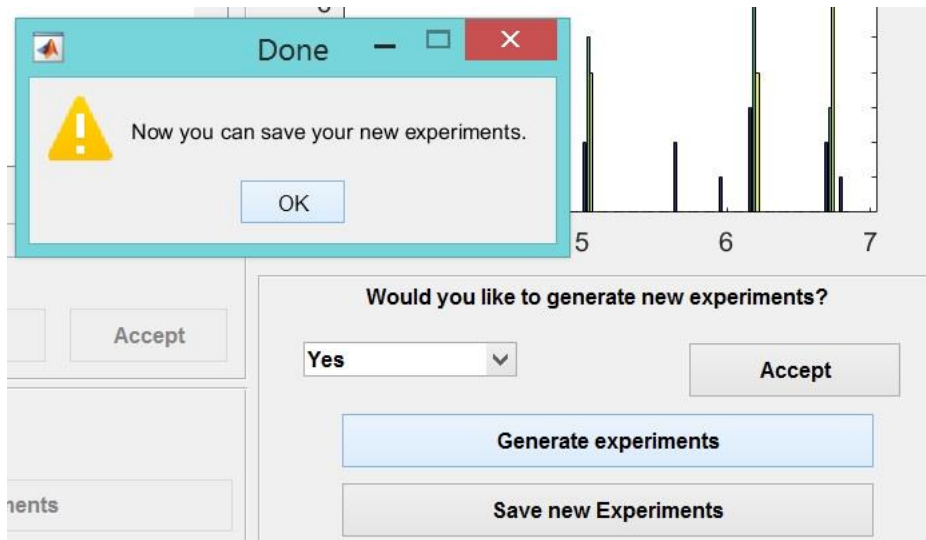


Load parameters	Load Experiments
Plot results	Save scores
Profile ▼	Save results

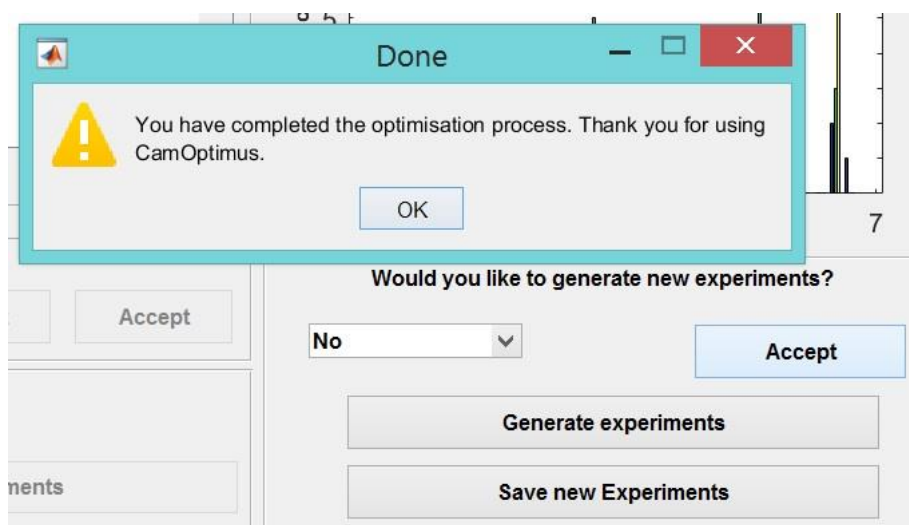
The individual non-normalised scores for each objective, which can be used as an indicator of the improvement of the system over time, are plotted in the first Figure with the title “Average scores for the selected objective in each generation”. The selection for individual objectives can be made from the “Objectives list” dialogue box. If replicate experiments are carried out, the mean or the median of the replicates can be provided for each individual experiment. An absolute frequency plot, entitled as “Population profiling for the selected factor”, displaying (i) the distribution of values employed by a given factor over the generations (by selecting “Profile” from the drop-down menu), and (ii) the distribution of values employed in the best performing fraction of the most recent generation (by selecting “Best” from the drop-down menu) are displayed when the “Plot results” command is prompted while a factor from the “Factors list” is selected. The absolute frequencies are binned to create the histograms. By selecting a factor from the list and clicking “Plot results” every time, the evolution of each individual factor can be monitored. The numerical values for the scores and the absolute frequency plots are saved in a separate spreadsheet by prompting “Save scores” and “Save results”, respectively.



The user then makes a decision on whether if they would like to proceed with a next generation of experiments, based on how satisfactory the convergence of the factors and the scores were. If “Yes” is accepted, the user is then allowed to “Generate experiments” and “Save new experiments”. For convenience, we suggest saving everything into the same spreadsheet files by allowing the files to be overwritten so that the results can be as concisely and completely presented as possible. Please note that the previous data entries will not be deleted by this command.



If the convergence is deemed sufficient by selecting “No” for the question “Would you like to generate new set of experiments?”, a pop-up message appears indicating the termination of the procedure.



The user is then ready to take their results to the next level to investigate their solution space.

B. SYMBOLIC REGRESSION

Once the Symbolic Regression option is selected in the main page, the following interactive dialogue appears:

The dialog box is titled "gene_symbolic_regression". It contains the following elements:

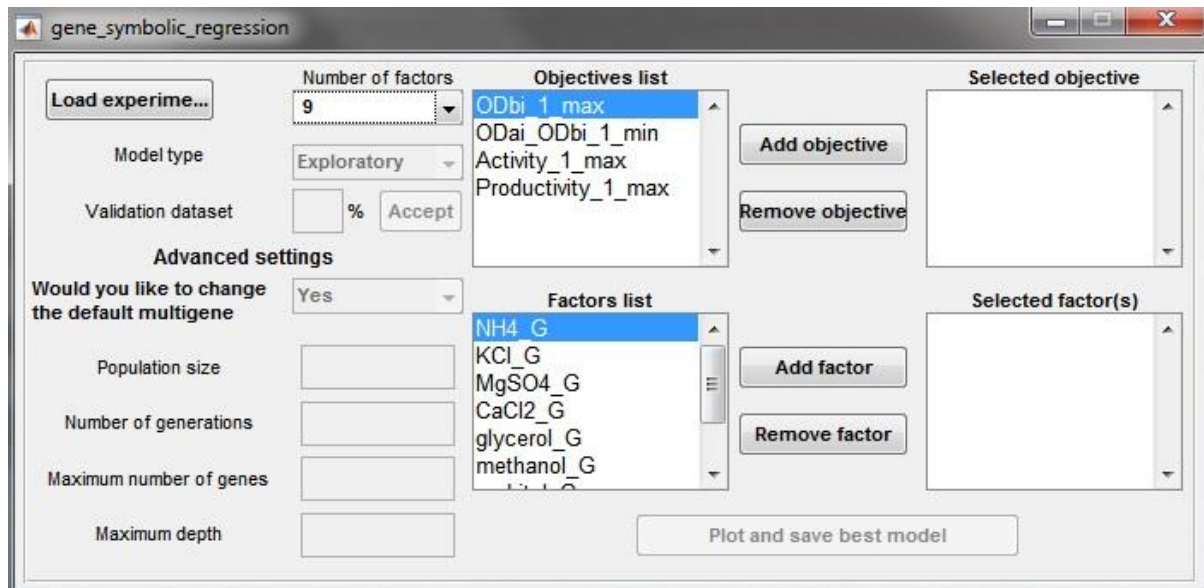
- Load experime...:** A button to load an experiment.
- Model type:** A dropdown menu currently set to "Exploratory".
- Validation dataset:** A text input field followed by a "%" sign and an "Accept" button.
- Advanced settings:** A section containing:
 - Would you like to change the default multigene:** A dropdown menu currently set to "Yes".
 - Population size:** A text input field.
 - Number of generations:** A text input field.
 - Maximum number of genes:** A text input field.
 - Maximum depth:** A text input field.
- Objectives list:** A list box for selecting objectives, with "Add objective" and "Remove objective" buttons to its right.
- Factors list:** A list box for selecting factors, with "Add factor" and "Remove factor" buttons to its right.
- Selected objective:** A list box for the selected objective.
- Selected factor(s):** A list box for the selected factor(s).
- Plot and save best model:** A button at the bottom right.

As in Section A, the interactive interface allows actions to be taken in a given order so as to guide the user through the steps of the procedure.

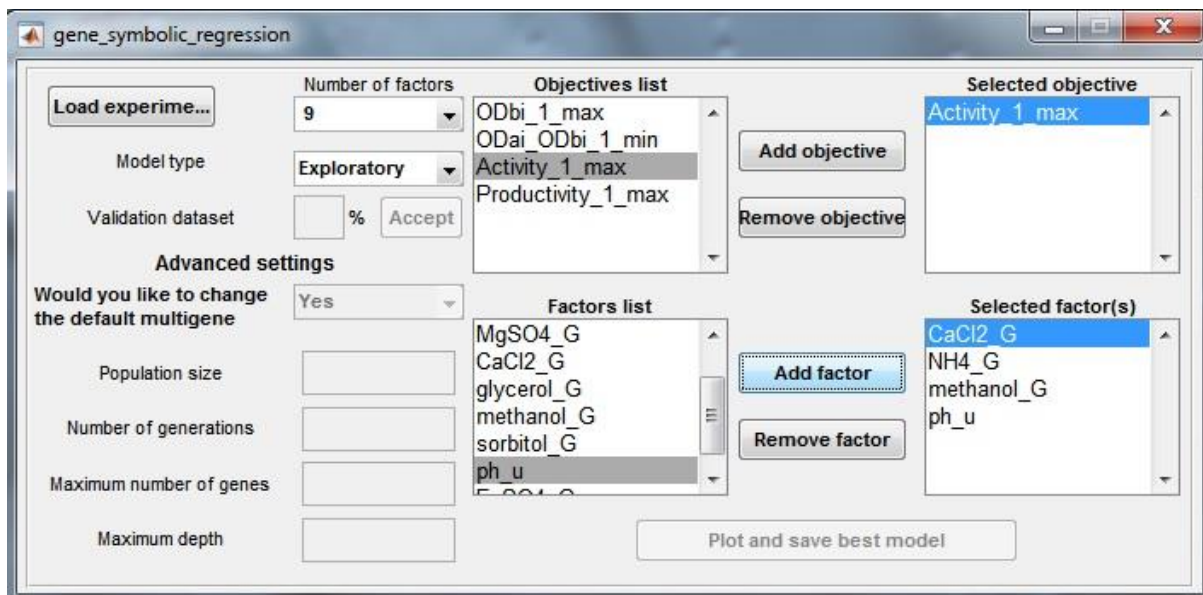
1. Loading the data and model settings

The regression analysis is designed to follow the optimisation protocol employed in Section A, and therefore the structure is designed to make use of the data generated in the earlier stages. Clicking "Load experiments" will prompt the user to locate the "Experiments" spreadsheet generated in Section A. However, the tool will accept any other spreadsheet, which was prepared in a similar format to that generated by the Genetic Algorithm Section of the Tool. The first column of the spreadsheet is recognised as the identifier column for the experiments, and by selecting the number of columns dedicated to the factors under

investigation, the user classifies the columns into the factors and the objectives, which then appear in their cognate boxes.

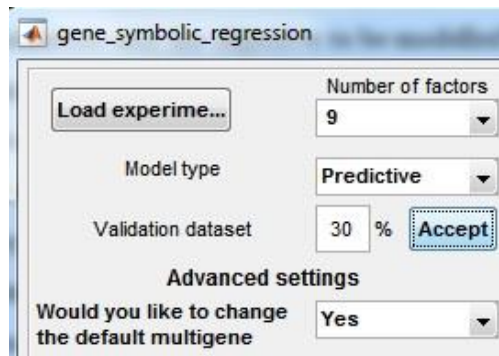


The user then selects which parameter(s) would be employed in describing which objective by the constructed model. For this purpose, one objective is selected from the “Objectives list” and highlighted in the “Selected objective” box by clicking the “Add objective” button. The objective under investigation can be changed by first selecting and removing the selected objective and then selecting a new one from the list and adding it as the selected objective. Similarly, the factors that contribute to the selected objective (outcome) are selected one by one from the “Factors list” and added to the “Selected factor(s)” menu using the “Add factor” button. One or more factors may be changed by making use of the “Remove factor” command. An example setup where the effect of pH as well as the amount of methanol, ammonium and calcium chloride on enzyme activity would be investigated is shown below:



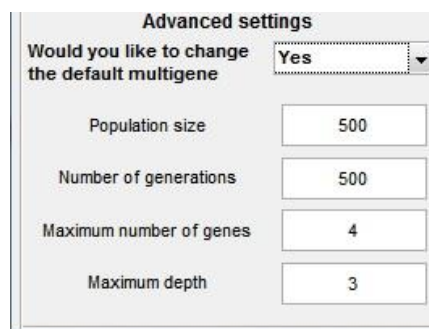
2. Identifying model type

Once the model parameters and the objective to be modelled are determined, the user then has to decide on the purpose of the model to be constructed. The tool is suitable for constructing both descriptive models and predictive models. The exploratory model option employs as much of the experimental data made available as possible to describe the solution space and through evolutionary approaches, attempts to reach an optimal model structure as well as regression coefficients in order to best fit a function to the available data. The goodness-of-fit of the model is described by how well the model fits the experimental data. The predictive model option retains a pre-defined percentage of the complete dataset as the validation dataset in a completely randomised manner and employs the remaining fraction (the training dataset) for constructing the model with the best fit available. This time, the goodness-of-fit of the model is described by how well the model constructed using the training dataset fits the validation dataset and this is an indicator of its predictive success, by definition. If a predictive model will be constructed, a given percentage of the data may be excluded from the training exercise by entering a value in the % box and prompting the “Accept” command. The default value suggested for the user is 25%. The figure below shows the setup for a predictive model with 30% of the data reserved as the validation set:



3. Employing advanced settings

Following the selection of the type of model to be constructed, the advanced settings for the symbolic regression can be adjusted by prompting the user to accept or change the default settings provided in the software. Although some suggested default values for the population size, number of generations, maximum number of genes and the maximum depth are provided in the tool itself, we urge the user to change these default settings in order to construct models with improved goodness-of-fit. Increasing population size and the number of generations improves the model fitness, whereas increasing the maximum number of genes and the maximum depth increases the complexity of the constructed model, and may also contribute to the fitness of the model.

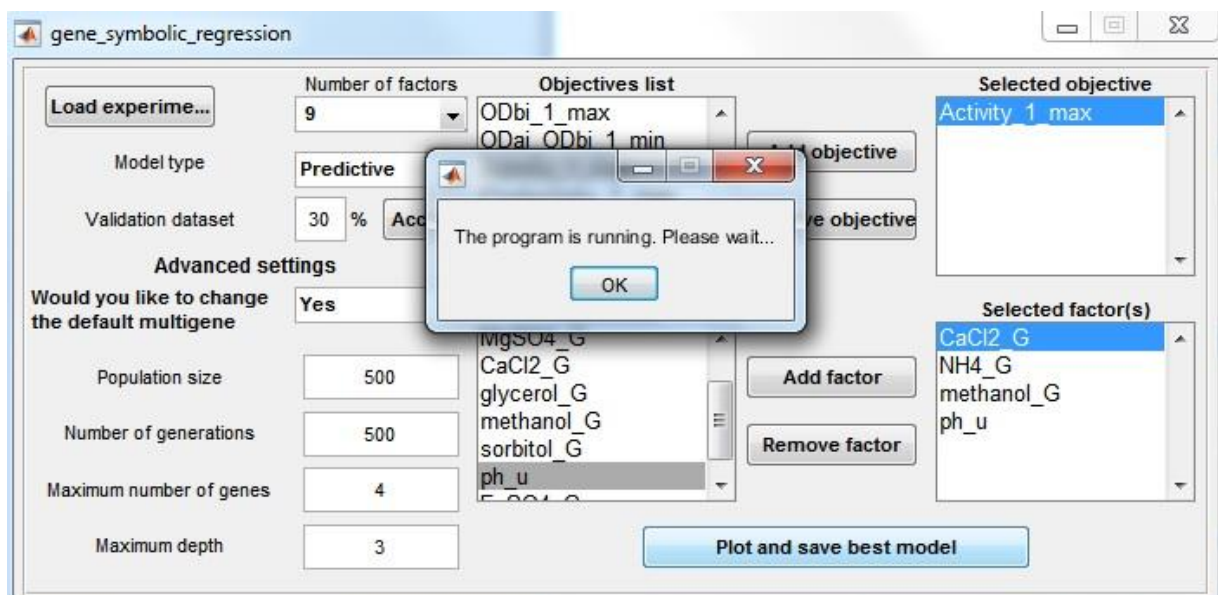


It is worthwhile to note that the parameter values provided here should be considered as independent of those discussed in Section A. The concepts and the parameters employed in describing these concepts are similar for genetic algorithms and other approaches including

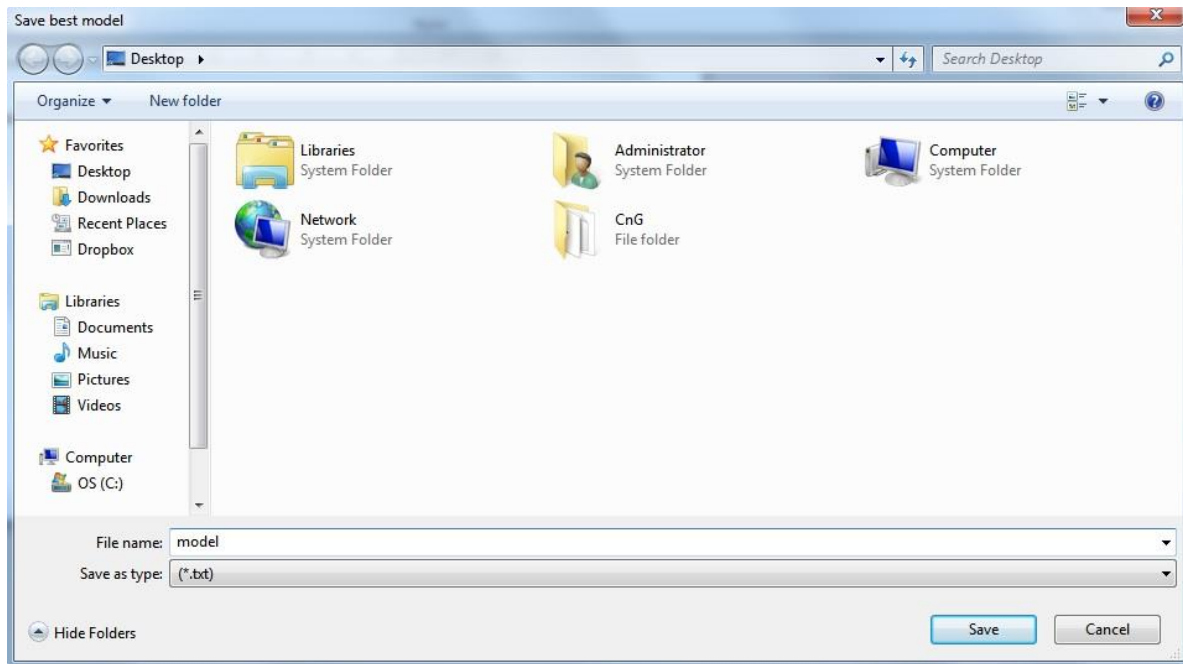
but not limited to symbolic regression, which employs genetic programming, and therefore should be evaluated within their own context.

4. Model construction

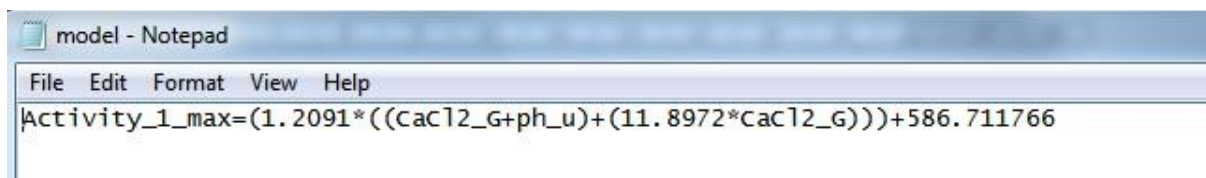
Once these settings are determined, the tool is then ready to construct the model by prompting the “Plot and save best model” command. Depending on the parameters selected, the program may take a while to run and you may receive the following message:



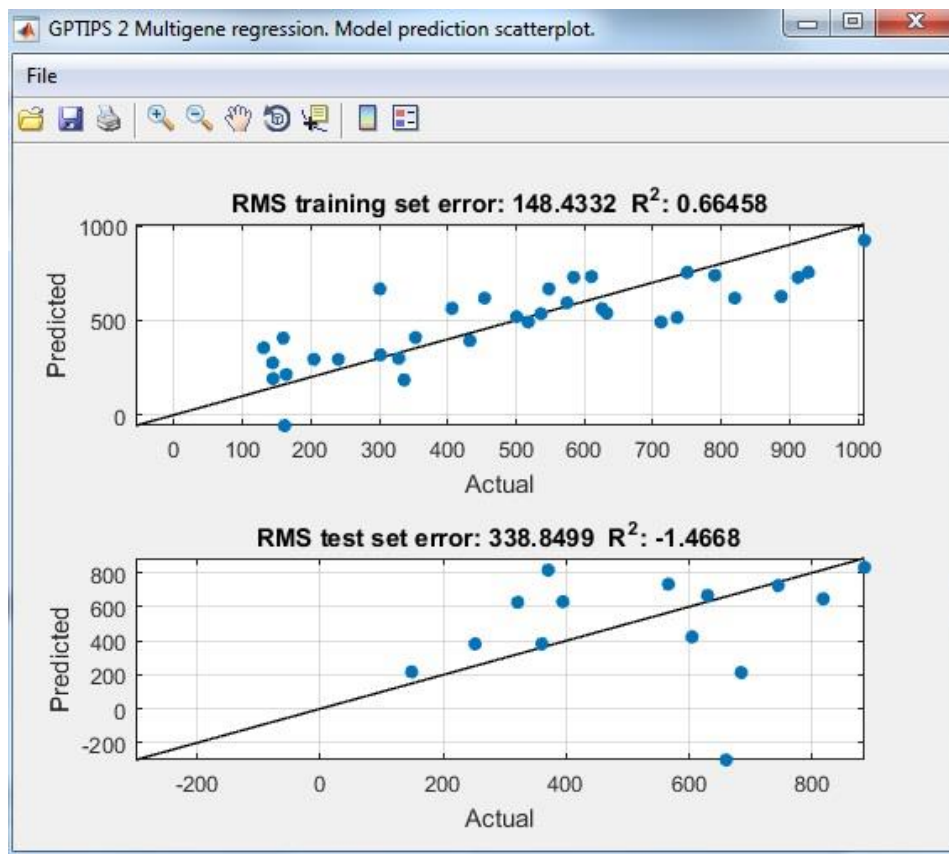
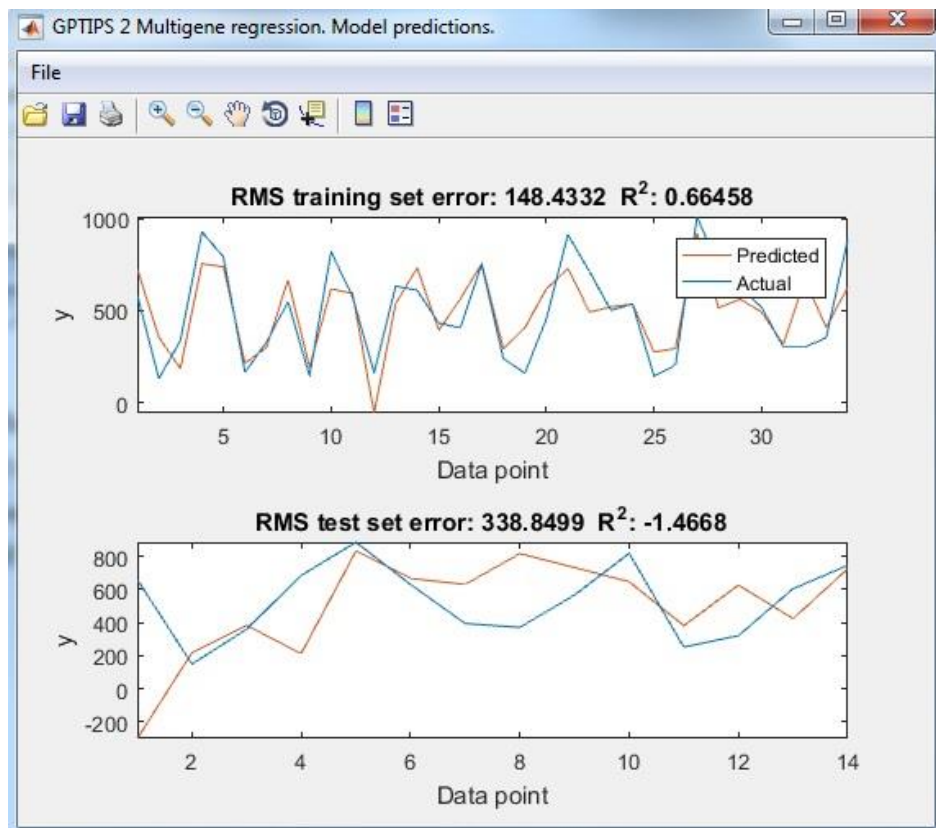
Four windows appear once a model is constructed; (a) The user is prompted to save the best fitting model in a text file (.txt extension):



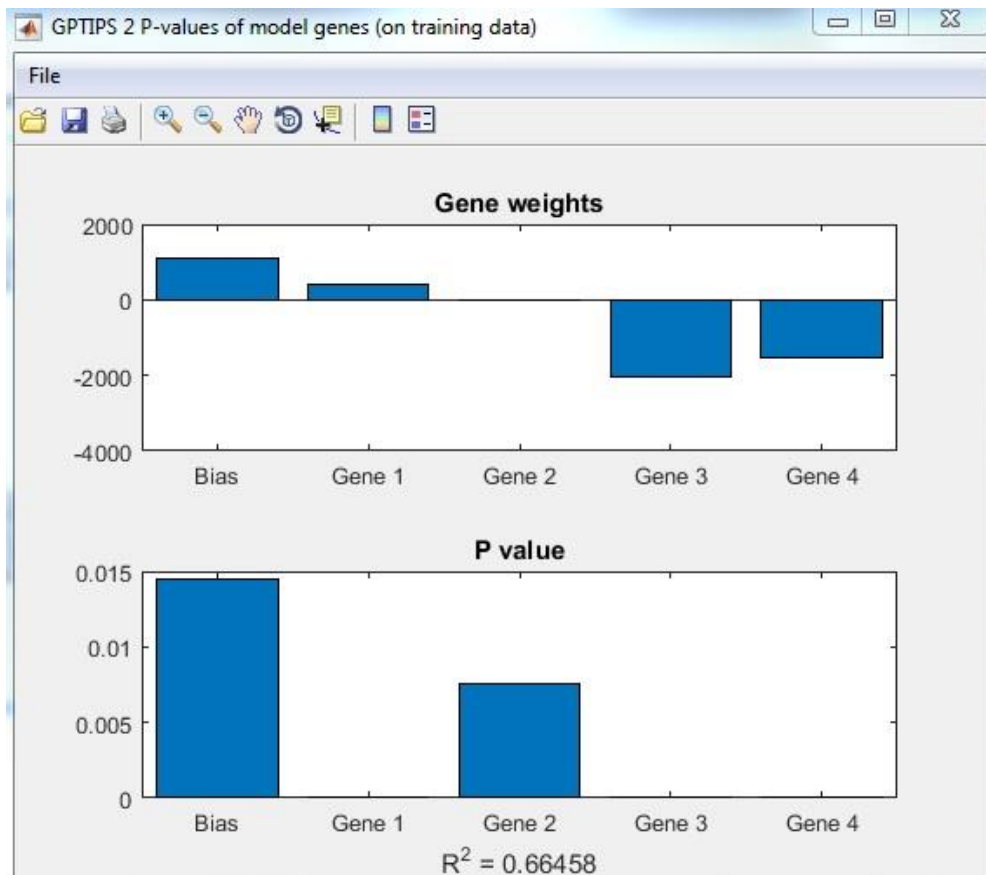
The saved document has the model structure recorded for any possible further analyses (e. sensitivity analysis):



(b) and (c) The user is provided with the model prediction and actual data plots for each data point (b) and the scatter plots – prediction vs actual data (c) for the training (upper plot) and the test (lower plot) sets along with the room mean squared (RMS) error and the regression coefficient (R^2) values for each set:



(d) The final window provides information on the significance of each factor (i.e. gene – represented in the order selected in the Tool interface). The first plot in the window shows the relative significance of each factor in the model and the second plot provides information on the significance of the information provided in the upper plot.



...

End of Manual