GPdotNET v4.0 User Guide

dr. sc. Bahrudin Hrnjica
2015

Software Disclaimer

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

1. Modeling with GPdotNET

GPdotNET is C# open source artificial intelligence tool for applying Genetic Algorithm and Artificial Neural Networks in modeling, prediction, optimization and pattern recognitions. With GPdotNET you can solve various engineering problems from classic regression and approximation to linear programming transportation and location problems and other machine learning based problems. By providing the learning algorithms GPdotNET uses a data of the research or experimental measures to learn about the problem. The results of learning algorithms are analytical models which can describe or predict the state of the problem, or can recognize the pattern. GPdotNET is very easy to use, even if you have no deep knowledge of GA, GP or ANN, you can apply those methods in finding solutions. The project can be used in modeling any kind of engineering process, which can be described with discrete data, as well as in education during teaching students about evolutionary methods, mainly GP and GA, as well as machine learning mainly Artificial Neural Networks.
The typical process of modelling with GPdotNET can be described in 5 steps.

1. **Choosing the Solver Type**: The first step is choosing the type of the solver. Which solver you will use depends on your intention what you want to do. For example if you want to make model for your experimental measurement you have several options which depend of your experimental data and the method you want to use. In GPdotNET you can use Genetic Programming or Neural Nets for modelling and prediction experimental data. But this is not strictly separate as may look on the flowchart below. That means that you can user Neural Networks for prediction, but training algorithm can be based on Genetic Algorithm or Particle Swarm Optimization or Back Propagation algorithm.
2. **Loading Experimental Data**: GPdotNET uses powerful tool for importing your experimental data regardless of the type of data. You can import your numerical, binary or classification data. GPdotNET can automatically define classes, or format numerical data with floating or comma separated decimal values. More info can be find in Section 2.
3. **Setting Learning Parameters**. After data is loaded and prepared successfully, you have to set parameters for the selected method. GPdotNET providers various parameters for each method, so you can set parameters which can provides and generates best output model.
4. **Searching for the solution**: GPdotNET provides visualization of the searching solution so you can visually monitor how GPdotNET finds better solution as increasing the iteration number. If you provide data for testing calculated model, you can also see simulation of prediction.
5. **Saving and exporting the results**: GPdotNET provides several options you can choose while exporting your solution. You can export your solution in Excel or text file, as well as in Wolfram Mathematica or R programming languages.

As can would be seen, working in GPdotNET follows the same procedures regardless of the problem type. That means you have the same set of steps when modelling with Genetic Programming or Neural Networks. In fact GPdotNET contains the same set of input dialogs when you try to solve Traveling Salesman Problem with Genetic Algorithm or if you try to solve handwriting recognition by using Backpropagation Neural Networks. All learning algorithms within GPdotNET share the same UI.

The picture below shows the flowchart of the modelling in GPdotNET. The five steps described previously are depicted in the graphical forms surrounded with Start and Stop elements.

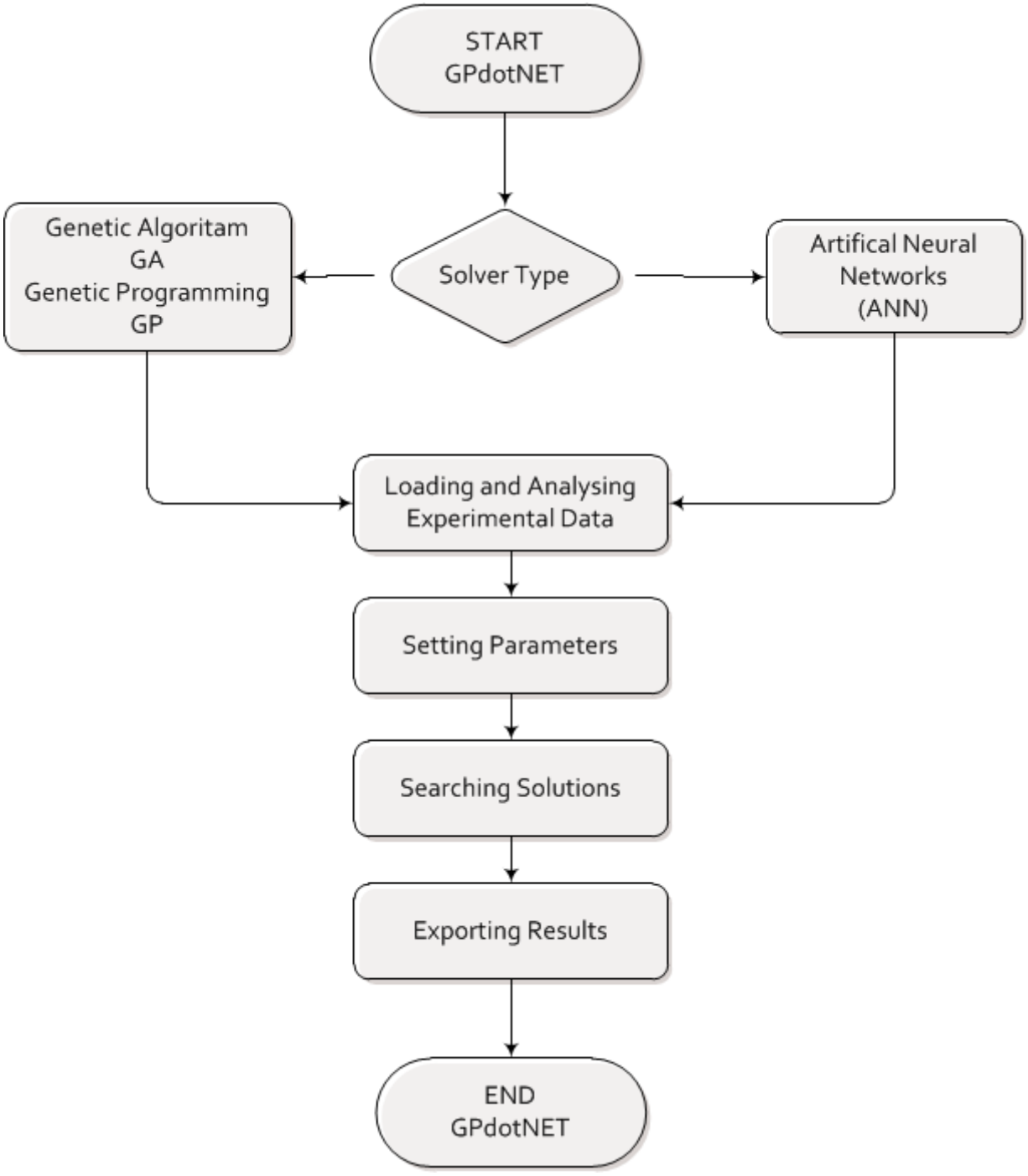


Figure 1.1. Modelling in GPdotNET

Besides parameters specific to learning algorithm, GPdotNET provides set of parameters which control the way of how iteration process should terminates as well as how iteration process should be processed by means of parallelization to use the multicore processors. During the problem searching GPdotNET records the history, so you can see when the best solution is found, how much time pass since last iteration process start, or how much time is remain to finish currently running iteration process.

Due to the fact that GP is the method which requires lot of processing time, GPdotNET provides parallelization, which speed up the process of searching. Enabling or disabling the parallelization processing is just a click of the button.

GPdotNET Open source project

From developer point of view GPdotNET is .NET (Mono) application written in C# programming language which can run both on Windows and Linux based OS, or any OS which supports Mono framework. Project started in 2006 within postgraduate study for modeling and optimization with evolutionary algorithms. As open source project, GPdotNET is first published on November 5 2009 on codeplex.com. The project is licensed under GNU Library General Public License (LGPL). For information about license and other kind of copyright please see http://gpdotnet.codeplex.com/license. The project is hosted at http://gpdotnet.codeplex.com. Main place for all news, documentation and code changes is my blog site at http://bhrnjica.wordpress.com/gpdotnet.

How to citate GPdotNET

GPdotNET is used from all around the world, in scientific papers, journals, books, for diploma works, master thesis or Dissertations. It is free to use GPdotNET with proper citation. So if you want to use the GPdotNET you need the right way to citate the tool.

Use this citation example in your paper, book or other material:

[1] **B. I. Hrnjica**, *GPdotNET V4.0- artificial intelligence tool [Computer program]*, http://gpdotnet.codeplex.com, accessed {date}.

Or

[1] **Bahrudin I. Hrnjica**, *GPdotNET V4.0 – artificial intelligence tool [Computer program]*, http://gpdotnet.codeplex.com, accessed {date}.

2. GPdotNET UI

GPdotNET has very simple and intuitive UI without any unnecessary buttons or icons. All available commands are located on first screen. Start screen contains toolbar with main commands for handilng files, running searching algorithms and closing the application. Main part of the screen is Start Page which contains useful links for opening predefined samples of all supported problem and solver types. The best way in starting with GPdotNET is choosing the one of the predefined sample and look what contains in it.

Figure 2.1. Start Screen of the GPdotNETv4.0

Main parts of the Starts Screen are:

1. Title Bar
2. Application Tool Bar
3. Start Page
4. Application Status Bar

Title Bar

Title bar contains Icon, application name and system options on the right side. With system options you can Close, Maximize and Minimize application. In fact this is standard windows system options.

Application Tool Bar

Main Toolbar – exposes main commands in GPdotNET. The commands are grouped in to 4 major groups.

1. Model – gathers commands for manipulation of GPdotNET model file. There are options for Create, Open, Save, Export and Save as GPdotNET model file, as well as Close currently opened model. Those commands are self-explained.
2. Modeling contains three commands for Run, Stop and Optimize models. Commands are enabled or disabled automatically, whenever there is a possibility the user can achieve logic action. For example while GPdotNET is running, user cannot press Run button, because there is no sense to press this button. In this case Stop button is enabled. Optimize button is available when GPdotNET is ready for optimization.
3. Common group contains option for common usage: Info to show basic information and Copyright of the application, and Exit option to close the application.

GPdotNET Start Page

If you are new to GPdotNET, there is no better starting point that Start Page. With Star Page you can try one of the predefined and recalculated samples. In fact start page contains all information you need to begin using GPdotNET. From recalculated samples to links for documentation and other useful information about GPdotNET can be found here. Start Page contains Samples which are split by problem and solver types.

1. Modeling, Prediction and Optimization
    a. Regression and Approximation
    b. Classification
    c. Time Series modelling
2. Linear Programming
    a. Traveling Salesman Problem
    b. Location Problem
    c. Transportation Problem
3. Optimization Analytic Function

Blue links represent the GPdotNET models with Genetic Algorithm, and the greens are based on Neural Networks.

3. Creating your first GPdotNET Model

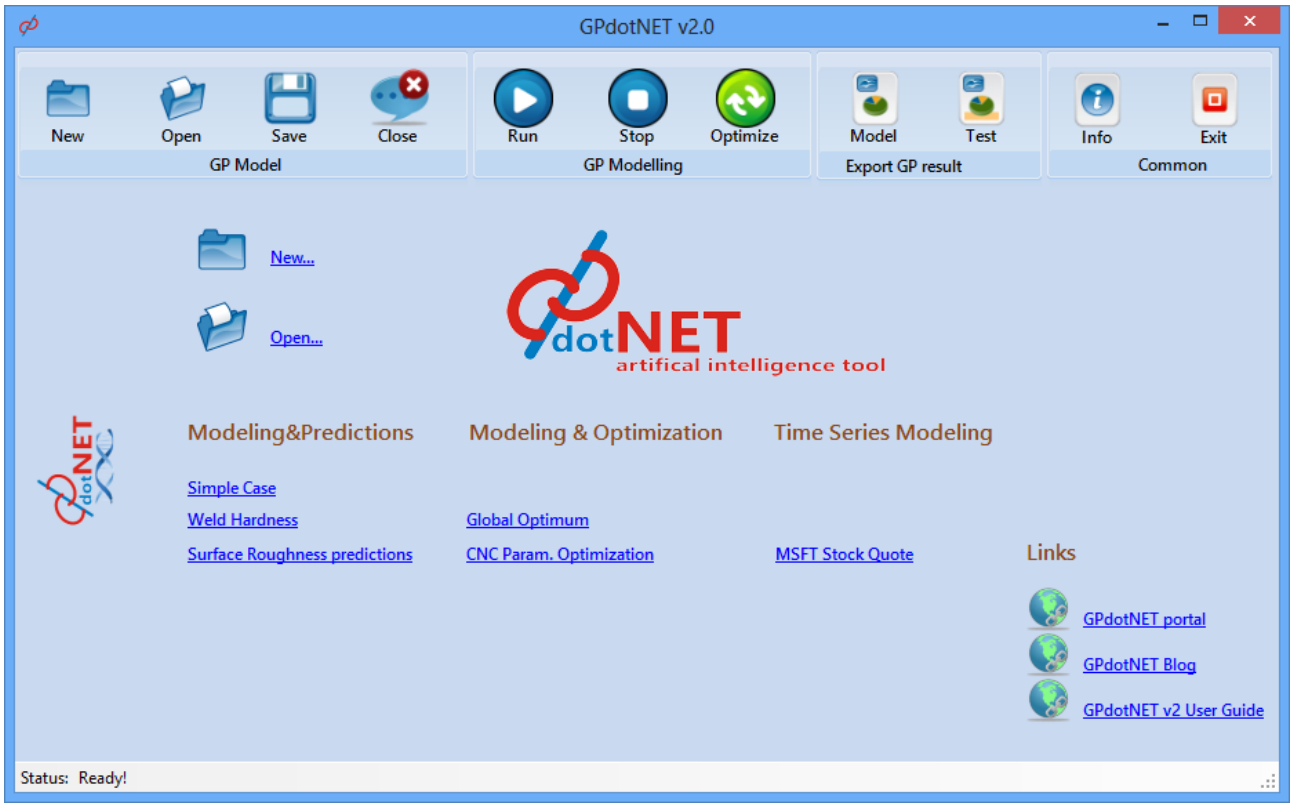


Figure 1: GPdotNET v2 Start Screen

The project is hosted at http://gpdotnet.codeplex.com for Windows users, as well as http://code.google.com/p/gpdotnet for Linux users. Main place for all news, documentation and code changes is my blog site at http://bhrnjica.wordpress.com/gpdotnet.

Note: If you have never heard about GP and GA, recommendation for getting basic information about GP is http://en.wikipedia.org/wiki/Genetic_programming. The wiki page also contains some links to other web sites about GP. For GA there is wiki page which contains a basic information about GA at this link http://en.wikipedia.org/wiki/Genetic_algorithm.

GPdotNET v2.0 supports the following types of modeling and optimizations:

1. **Model for Discrete Data** – modeling with/or without prediction of discrete data by using Symbolic Regression modeling with GP
2. **Model&Opt. for Discrete Data** - modeling with/or without prediction of discrete data by using Symbolic Regression with GP and Optimizing calculated GPdotNET model by using GA
3. **Model for Time Series** - Time Series modeling and prediction data by using Symbolic Regression with GP
4. **Optimization of Analytic Function** - optimization of analytic defined

function by using GA

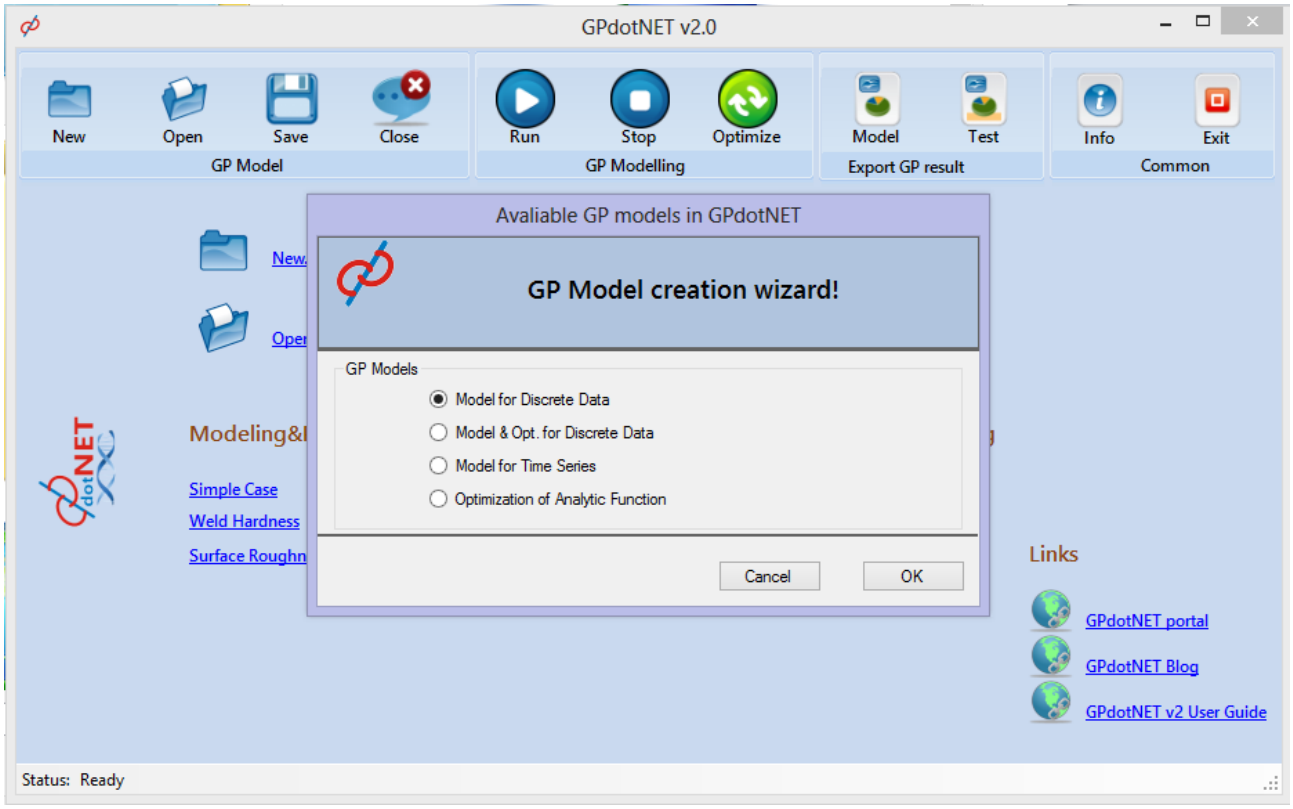


Figure 2: File New Dialog options in GPdotNET v2

4. Introduction to GPdotNET v2

With GPdotNET v2 I tried to solve some problems experienced during usage of the first version in modeling and optimization. Comparing to the previous version, this version is completely rewritten, and it is not compatible with version v1.0 in any segment, but general logic of applying evolutionary methods remains the same. General steps for settings parameters before running programs remain very similar. Parts of GP or GA are split in Tab pages so the user can have better understandings.

Note: During implementation of GPdotNET v2.0 I have tried to post as much information as I could, trying to provide all relevant information to users. This user guide mostly contains my blog posts, I was writing during implementations.

In the following text, it is listed the main features in GPdotNEt v2.0 which are new to version 2, as well as existing features.

Cross OS and Cross platform software (new in v2)

One of the main requirements for GPdotNET v2.0 is ability to run on multiple OSs, by using .NET and Mono Framework. So GPdotNET v2 can run on all OS where Mono is implemented. During the implementation every piece of code is tested against Mono. During implementation, when code was not compatible with Mono, it was replaced with the code implementation compatible with Mono. It can be said that the whole implementation was done using Visual Studio and MonoDevelop, working on Windows and Fedora 17. I didn't have much time to test GPdotNET on OS other that Windows 7 and Fedora 17, so every bug report would be appreciated.

Figure 3. GPdotNET v2 in MAC OS environment.

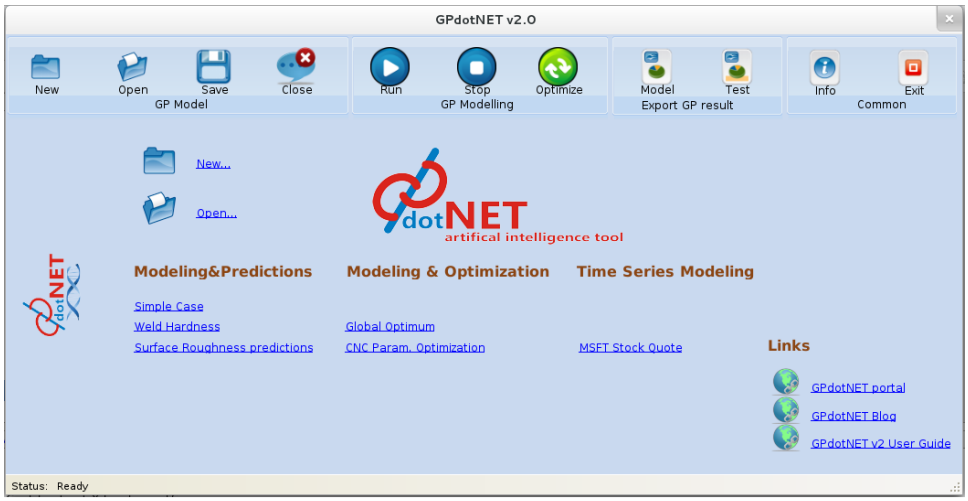


Figure 4. GPdotNET v2 in Fedora 17 OS environment

New text based file format *.gpa (new in v2)

GPdotNET V1.0 supported binary file format, and for large population size the file size was also big. On the other hand, with text file format you have possibility to modify file outside the GPdotNET. For example you can see whole population chromosomes, and see other data you are interesting in. You can also perform some manual modification if you like, by modifying training or testing data as well as parameters. In general manual modification file is not recommended.

Support for Excel and CSV export (new in v2)

Exporting in GPdotNET v2 is based on openXML file format, but there are some compatibility issues in Mono, so you cannot use Excel exporting in Mono. While you running GPdotNET v2 on Mono you can export data in CSV file format. This is only one feature which is not running in both Mono and .NET.

Optimization of GPModels (new in v2)

GPdotNET v2 can run optimization of calculated gpmodel. Optimization is very important for any engineering system. You can perform optimization after you perform modeling and got result. In fact you can run optimization and modeling as much as you want with only one constrains: You cannot run Optimization and Modeling at the same time.

Optimization of analytically defined function (new in v2)

GPdotNET v2 now supports optimization of any analytically defined function. You can define function in Tree expression designer, define constrains and perform optimization.

Support *.csv data file

GPdotNET support csv file format for loading training, testing and time series data. Common example of data file can be seen on the following picture.

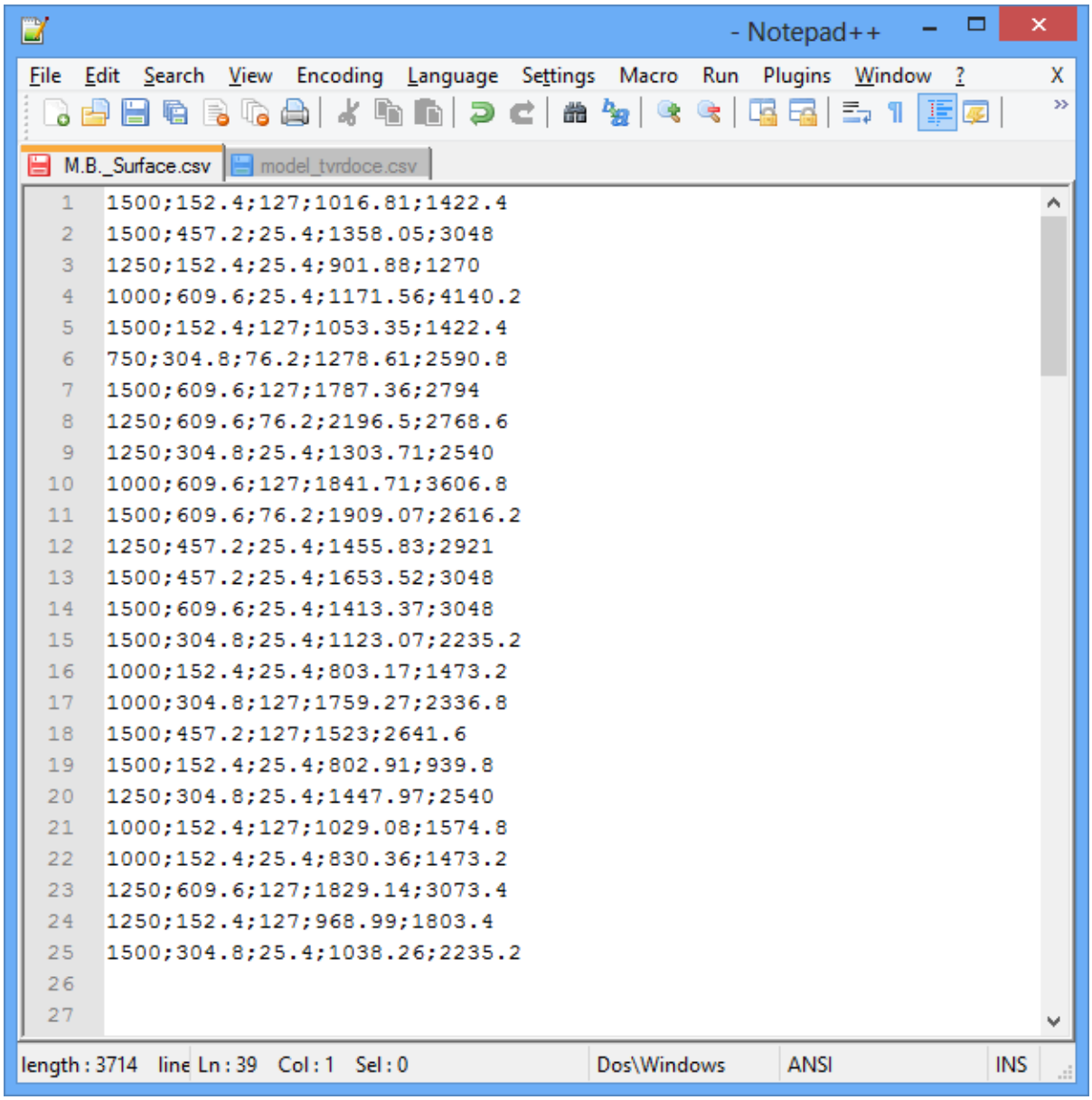
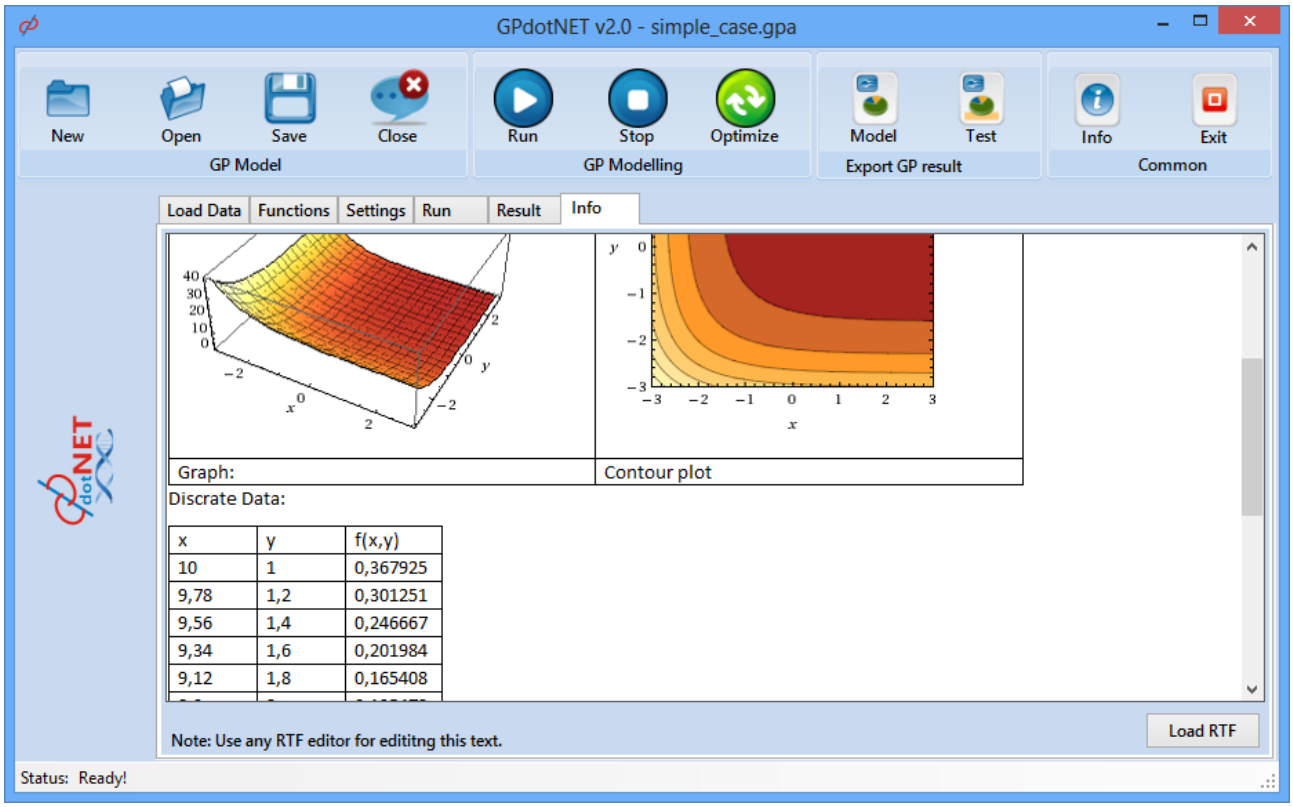


Figure 5. Sample of csv file

Regardless of user localization floating numbers must be written with decimal point. On this way we skip some complexity and localization issue seen in the previous version.
Columns are separated by semicolon, and rows with newline. The last column is always output variable. In case of Time Series, data file can contains only one column.

Info tab in Model (new in v2)

When you start with modeling and/or optimization a new Info Tab is created as well. Info tab contains rich edit control in which you can paste or load any rich text content from text to picture. On this way, you can attach textual information of you model.



New Look& Feel (new win V2)

Unlike previous version, GPdotNET v2 has new simplified GUI with only one big toolbar containing all available options. Commands are split in to 4 major groups: Model, Modeling, Export and Common. It is very simple and gives you all options directly on the screen. Run, Stop and Optimize commands are shifted to main toolbar, in order to give useability to stop or run programs from any tab page, not only from run page.

5. Working with GPdotNET v2

Picture below shows typical Start Screen of GPdotNET v2.0. Start screen can be divided in to several meaningful parts in order for better understanding UX of the GPdotNET v2.

Figure 6: GPdotNET Start Screen

Main parts of the Starts Screen are:

5. Title Bar
6. Application Tool Bar
7. Start Page
8. Application Status Bar

Title Bar

Title bar contains Icon, application name and system options on the right side. With system options you can Close, Maximize and Minimize application. In fact this is standard windows system options.

Application Tool Bar

Main Toolbar – exposes main commands in GPdotNET. The commands are grouped in to 4 major groups.

4. GPModel – gathers commands for manipulation of GPdotNET model file. There are options for Create, Open, Save and Save as GPdotNET model file, as well as Close currently opened model. Those commands are self-explained.
5. GP Modeling contains three commands for Run, Stop and Optimize GPModel. Commands are enabled or disabled automatically, whenever there is a possibility user can achieve logic action. For example while GP is running, user cannot press Run button, because there is no sense to press this button. In case Stop button is enabled. Optimize button is available when GPModel is ready for optimization.
6. Export GP Result contains options for exporting result to other format: Excel and CSV.
7. Common group contains option for common usage: Info to show basic information and Copyright of the application, and Exit option to close the application.

GPdotNET Start Page

If you are new to GPdotNET v2, there is no better starting point that Start Page. With Star Page you can try one of the predefined and recalculated samples. In fact start page contains all information you need to begin using GPdotNET. From recalculated samples to links for documentation all information about GPdotNET can be found here. Start Page contains Samples which are split in 3 major groups (see Fig. 2).

1. Modeling & predictions
2. Modeling & Optimization
3. Time Series Modeling

Note: This version doesn't support persisting Optimization of analytic function, so there are no samples for it.

The last group of links is links for documentation and User Guide.

Modeling & Predictions

Click on Simple Case link from Start Page. After some time GPdotNET Model is loaded similar picture shows below.

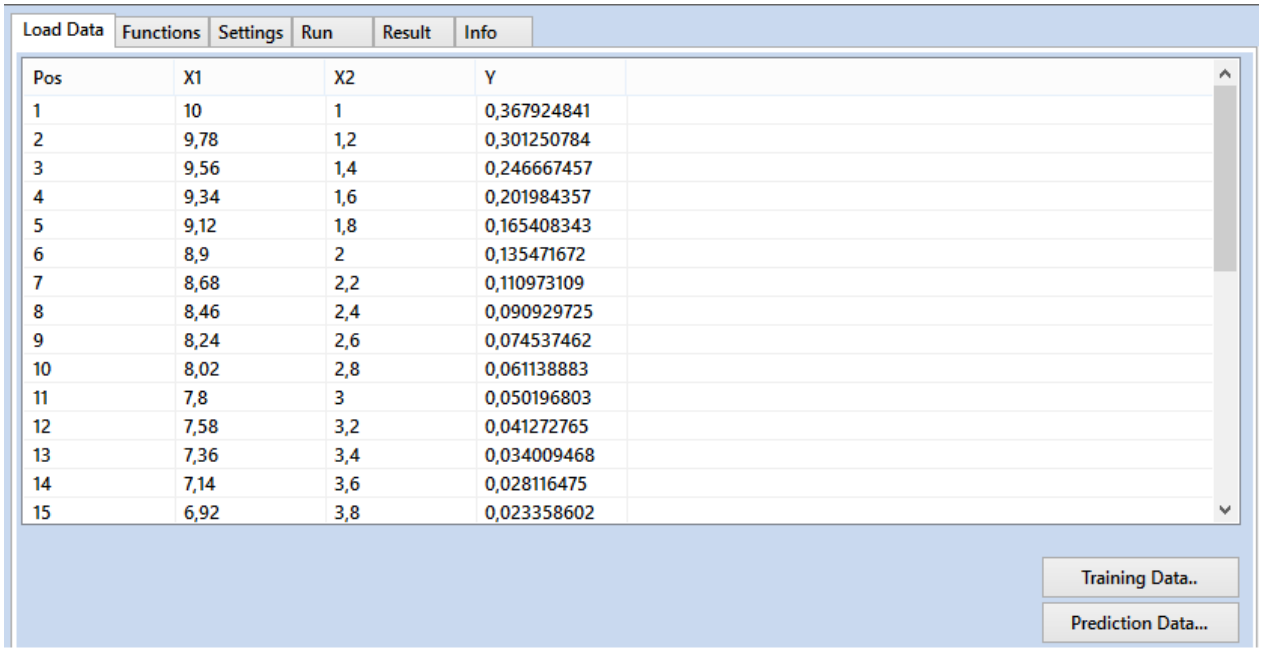


Figure 1: Load Data Tabe page in GPdotNET v2

There are several tab controls which separate information about model. GPdotNET v2 shows the following Tabs when you load sample from Modeling and Prediction. The same set of Tab control GPdotNET load when you select New, and choose the first option for Modeling.

Load Data Tab

The Load Data tab control (see Fig. 5) appears in every type of modeling. This is one of the main tabs. For every GP model you need a data, to train your model. When modeling common discrete data, Load Data Tab contains two buttons.

For loading training Data – with training data your GP model will be trained.

For loading testing data – with testing data your model will be tested, after model is calculated.

Note: You don't need to load testing data in order to make a GPdotNET model. Training data is used when you want to test calculated model.

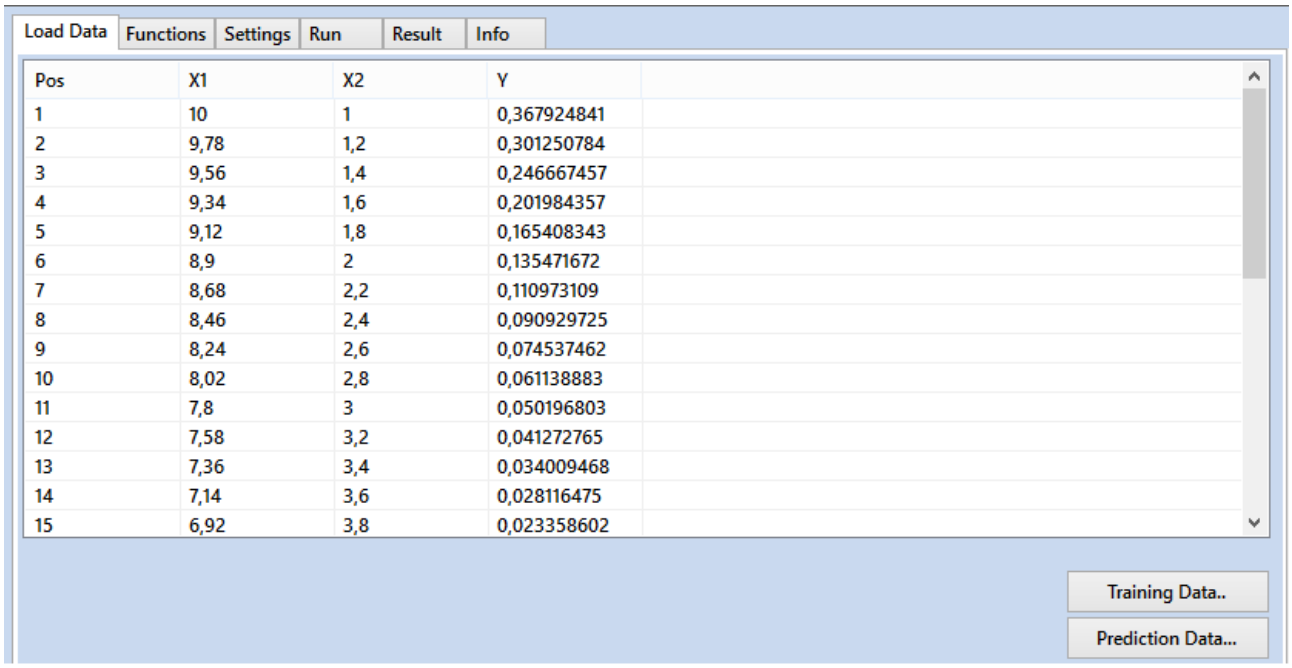


Figure 7: Load Data Tabe page in GPdotNET v2

Loading Time Series Data

With GpdotNET v2 you can perform modeling with TimeSeries dana as well. Run GpdotNET and click on MSFT Stock Quote. You opened Times series model. You can recognize thet Load dana is diferent that previous (see fig.)

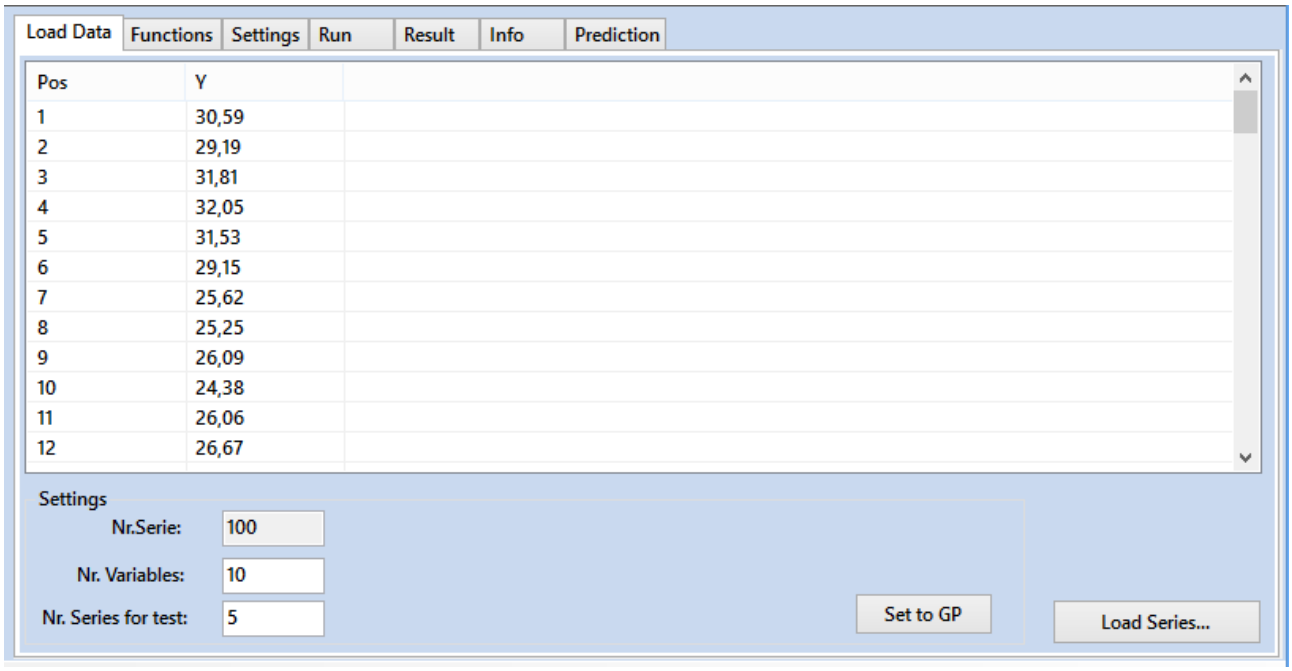
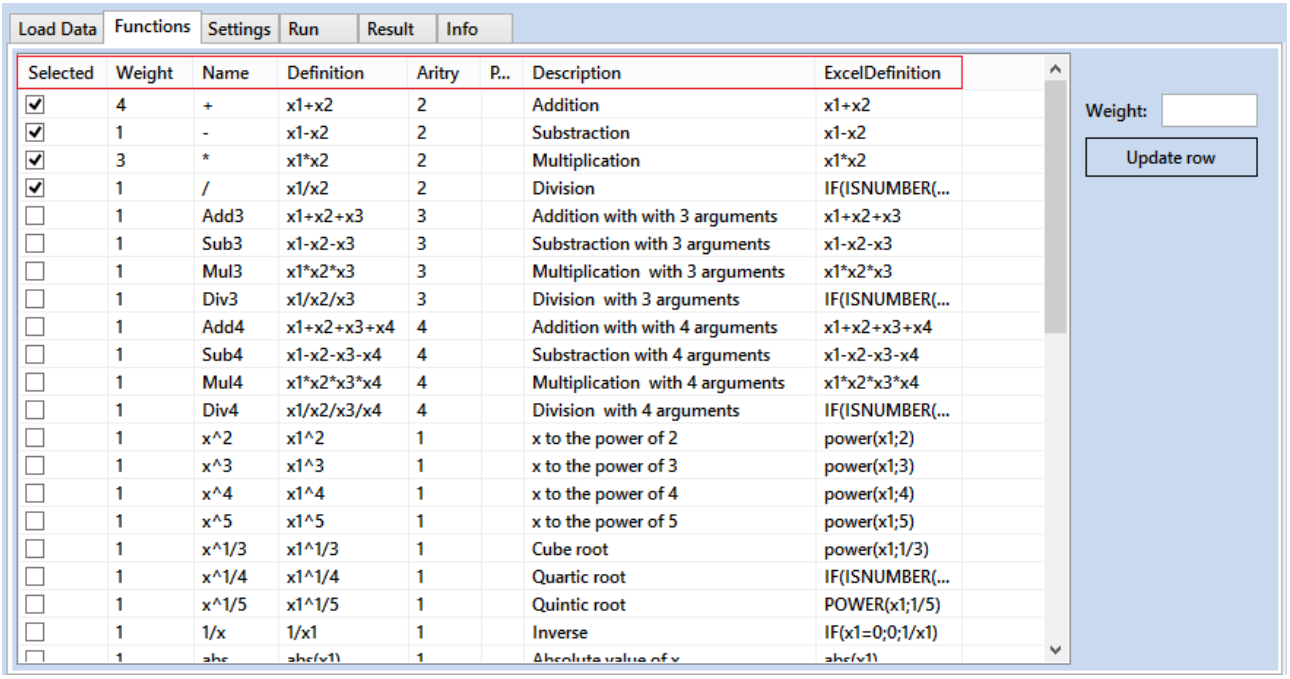


Figure 8: GPdotNEt Load Data tab in case of Time Series Modeling

Bottom part of Tab page, you can set number of variables to be as input variables, as well as if you want to define testing dana to test calculated model. After you defined variables and Test dana, in order to run, you need to press Setto GP button, that GPdotBET create training and testing dana.

Function Tab

Function tab contains all available function in GPdotNET. Function defined here will defined function set, from which model is constructed.



Selected	Weight	Name	Definition	Aritry	P...	Description	ExcelDefinition
☑	4	+	x1+x2	2		Addition	x1+x2
☑	1	-	x1-x2	2		Substraction	x1-x2
☑	3	*	x1*x2	2		Multiplication	x1*x2
☑	1	/	x1/x2	2		Division	IF(ISNUMBER(...
☐	1	Add3	x1+x2+x3	3		Addition with with 3 arguments	x1+x2+x3
☐	1	Sub3	x1-x2-x3	3		Substraction with 3 arguments	x1-x2-x3
☐	1	Mul3	x1*x2*x3	3		Multiplication with 3 arguments	x1*x2*x3
☐	1	Div3	x1/x2/x3	3		Division with 3 arguments	IF(ISNUMBER(...
☐	1	Add4	x1+x2+x3+x4	4		Addition with with 4 arguments	x1+x2+x3+x4
☐	1	Sub4	x1-x2-x3-x4	4		Substraction with 4 arguments	x1-x2-x3-x4
☐	1	Mul4	x1*x2*x3*x4	4		Multiplication with 4 arguments	x1*x2*x3*x4
☐	1	Div4	x1/x2/x3/x4	4		Division with 4 arguments	IF(ISNUMBER(...
☐	1	x^2	x1^2	1		x to the power of 2	power(x1;2)
☐	1	x^3	x1^3	1		x to the power of 3	power(x1;3)
☐	1	x^4	x1^4	1		x to the power of 4	power(x1;4)
☐	1	x^5	x1^5	1		x to the power of 5	power(x1;5)
☐	1	x^1/3	x1^1/3	1		Cube root	power(x1;1/3)
☐	1	x^1/4	x1^1/4	1		Quartic root	IF(ISNUMBER(...
☐	1	x^1/5	x1^1/5	1		Quintic root	POWER(x1;1/5)
☐	1	1/x	1/x1	1		Inverse	IF(x1=0;0;1/x1)
☐	1	abs	abs(x1)	1		Absolute value of x	abs(x1)

Figure 9: GPdotNET v2 Function Tab

There are two important columns which you can modify. Selected and Weight columns. When you want that certain function will be included in Function Set you need to check it. Weight column describe selection probability. From the picture above, + function has Weight=4, that means it has 3 times greater probability to be chosen, than subtraction function which has weight=1. On this way we can influence on probability of certain function in Function Set.

Changing Weight of the function

If you want to change weight of certain function do the following:

1. Click on certain function
2. Enter new weight value in text box on right side
3. Press button **Update row**.

Settings Tab

With Settings tab you can define parameters of genetic programming. GP parameters are Self-explained. If you don't know how to setup parameter, just live default values. It is suitable for most of problems.

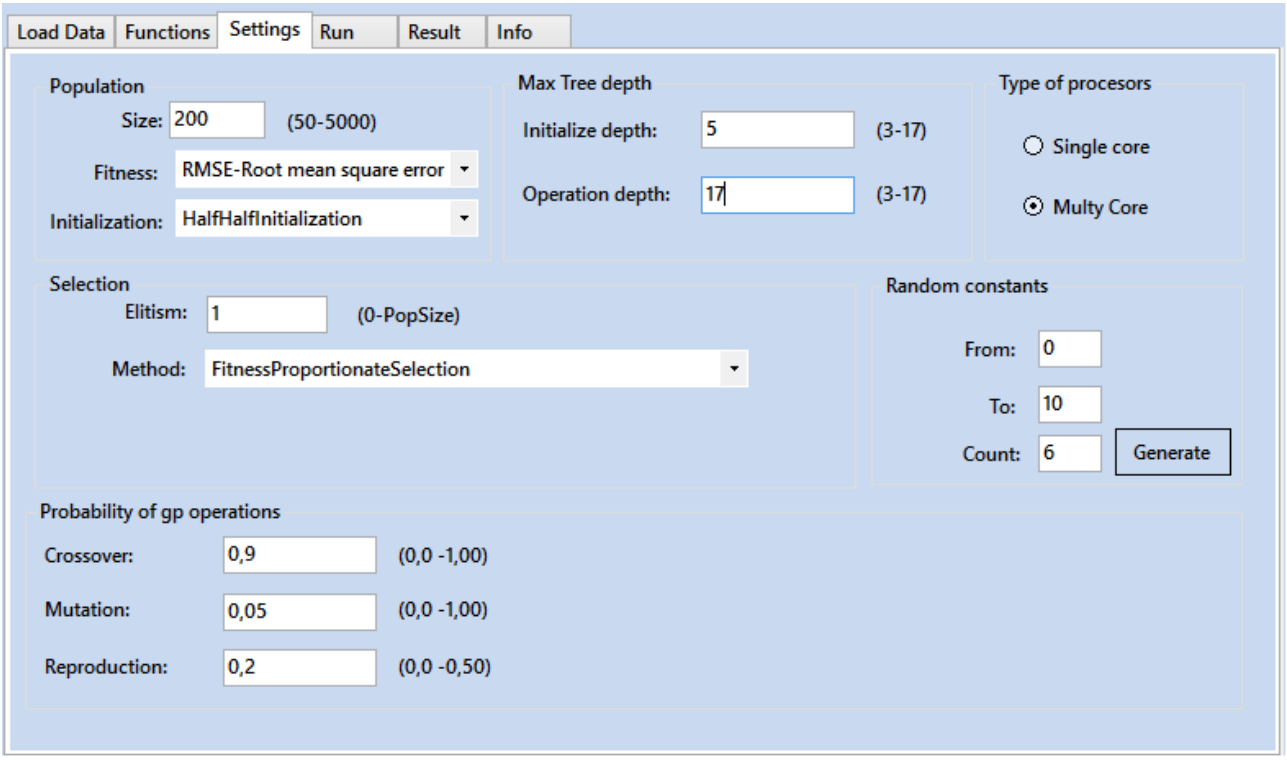


Figure 10: Setting Tab in GpdotNET v2

Settings tab also contains option to enable parallel processing of some certain GP calculation. So if you PC has more that one processor you can enable this kind of processing.

Run Tab

With Run Tab you control of GP modelling simulation, as well as defininf Termination criuteria of GP run. Rub Tab contains two Chart controls for simulation Fitness values, and GP Model during evolution of the program.

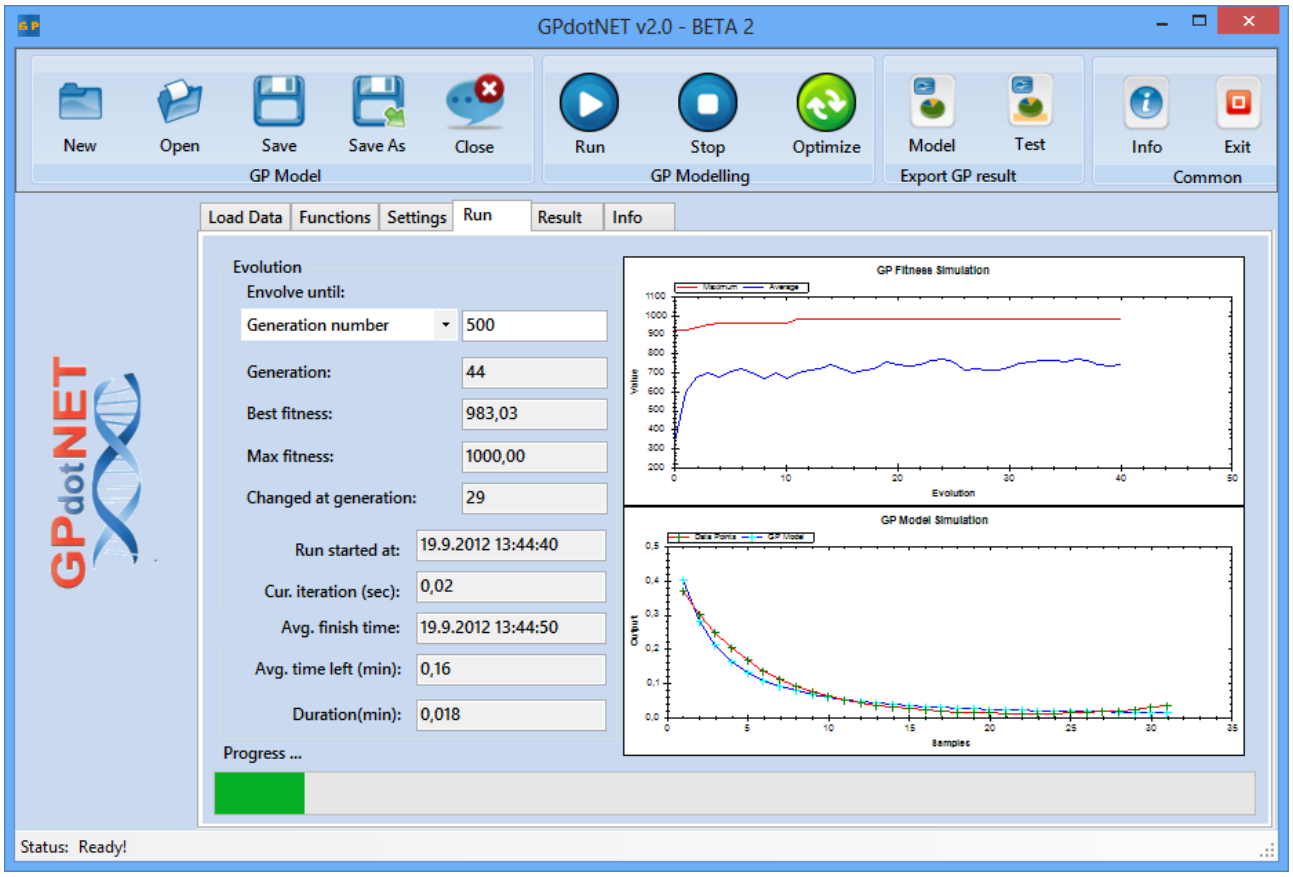


Figure 11: GPdotNET v2 Run Page

By using GP Modeling controls from the application toolbar you can control simulation and program running. In general there are 3 way to define termination criteria.

1. Number of evolution – when you chose the Generation number from the Combo Box
2. Fitness Exceed Value – when you choose ***Fitness*** **>=**from ComboBox.
3. If you click on Stop toolbar button during program execution

Setting Termination Criteria

Termination criteria can be set, when you coose one of the two predefined option in ComboBox. After you choose ComboBox option, you have to specifies value in edit box.

You can change termination criteria whenevery you want except during the programm execution while the controls are diabled for editing.

Result Tab

Result tab show current best solution GPdotNET calculated. The main space of Result tab occupy Tree Draw expression which show best solution in expression tree. Below tree

expression you can find solution in analytical form.

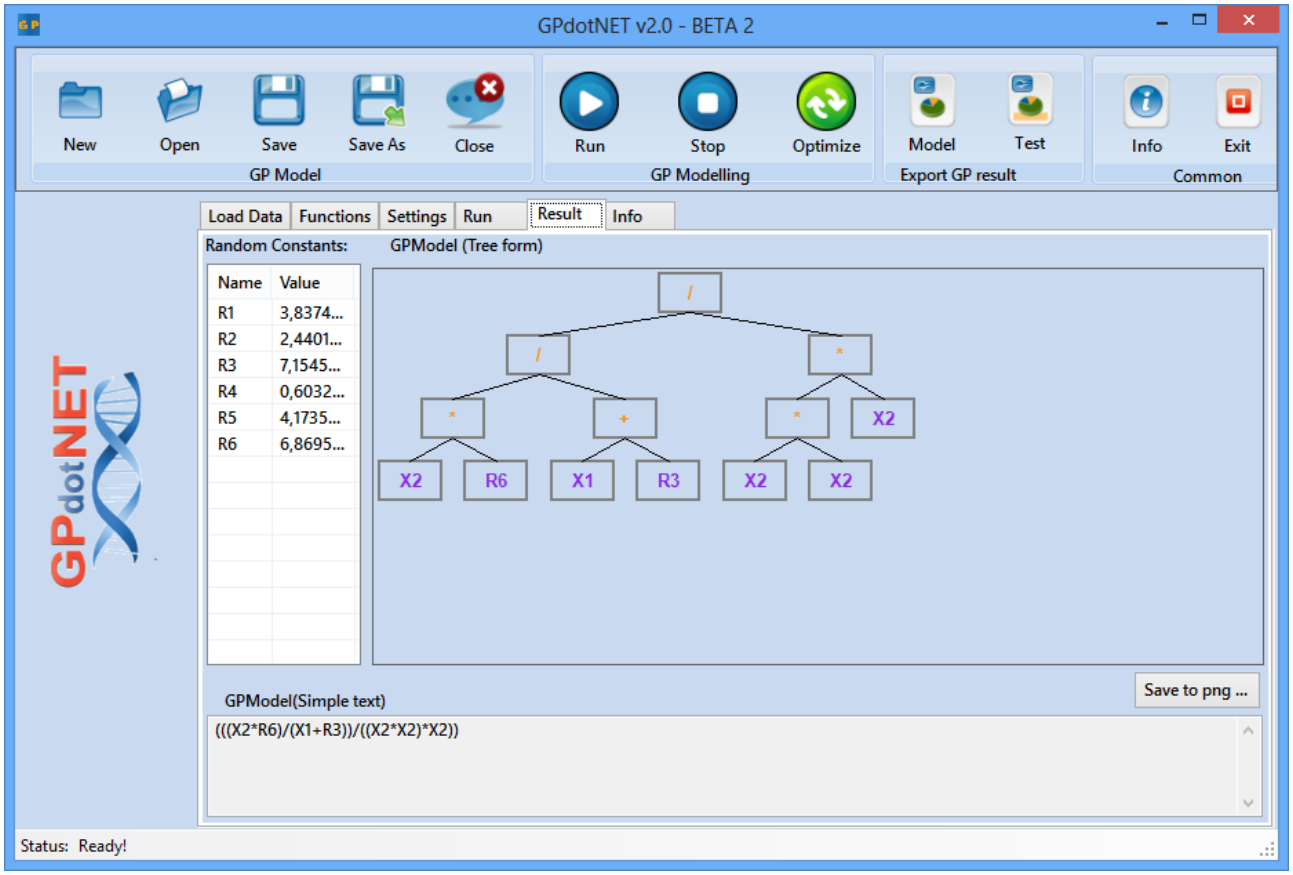


Figure 12: GPdotNET v2 Result Tab

Expression tree can also be saved in png image format.

Predition Tab

Perdition Tab is shown when you load Testing Data. In any other case Prediction Tab will not be shown in GPdotNET environment. Picture 9 shows Prediction Tab, which contains Table and Chart of predicted data calculated with the current solution.

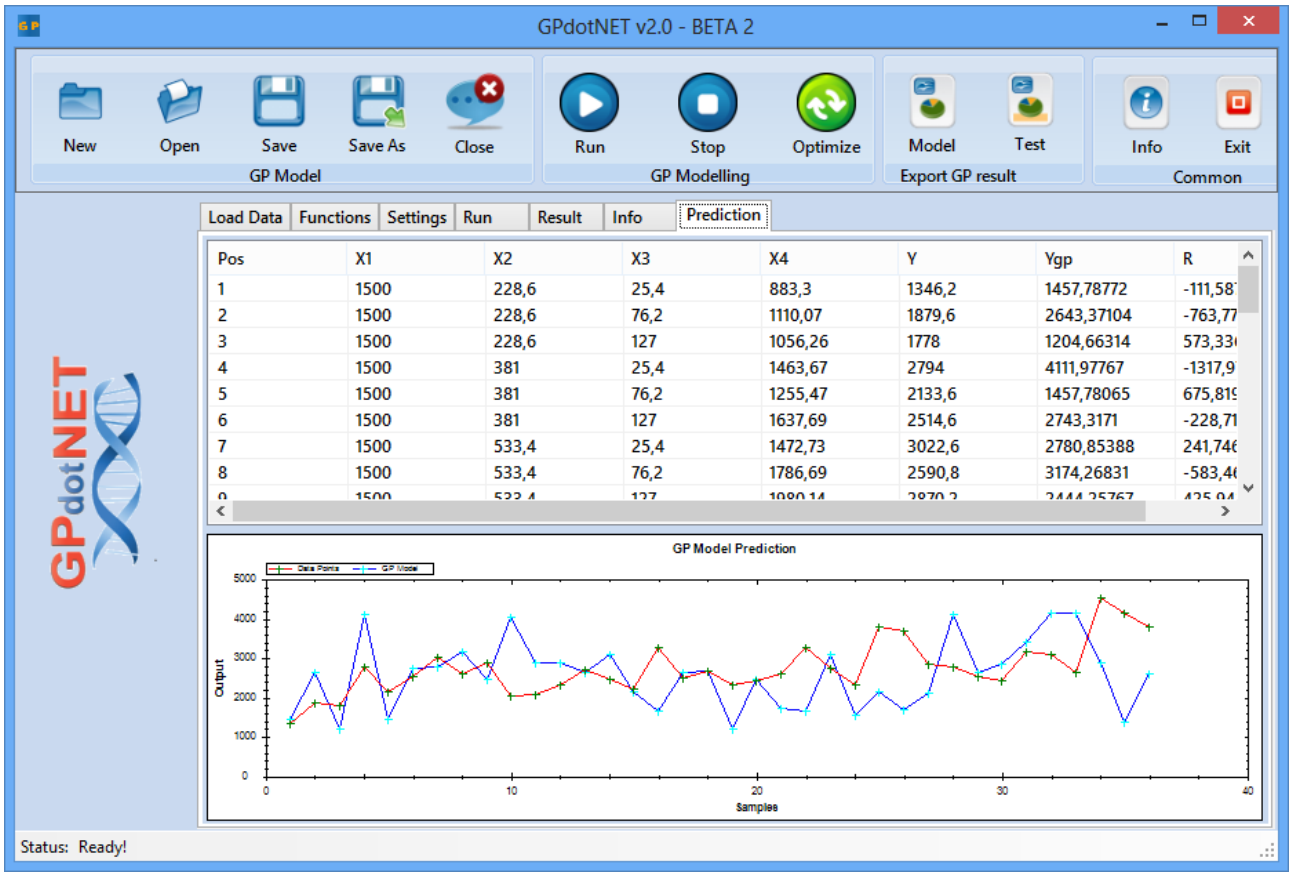


Figure 13: GPdotNET v2 Predition Tab

Info Tab

Info Tab is usefull when you want to attach some information about GpdotMET mode. Info Tab contains Rich Edit control which you can load any rtf file format. By choosing Laod rtf file, you can load rtf file from disk.

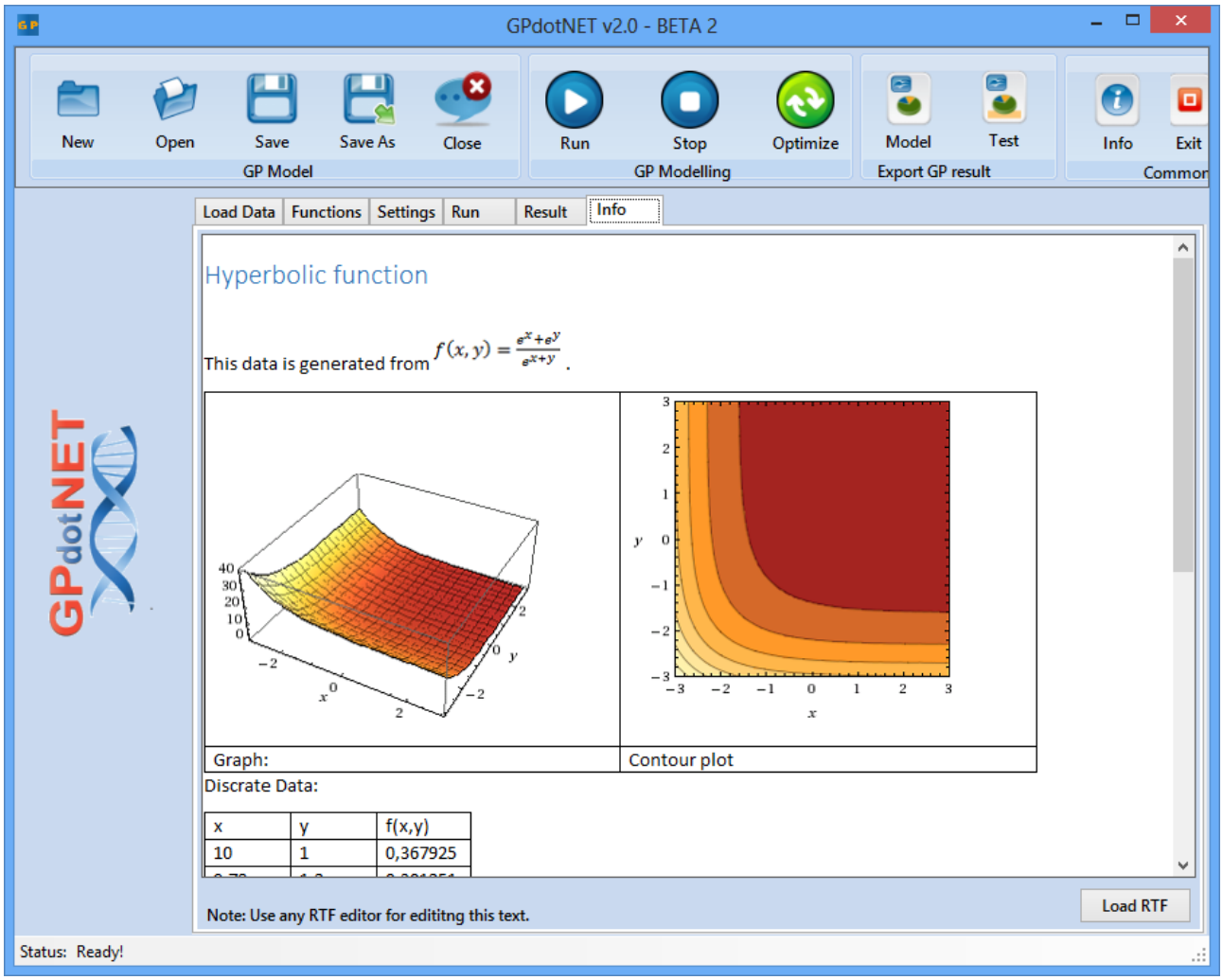


Figure 14: GPdotNET v2 Info Tab

Optimize Model Tab

When you choose Modeling& Optimization additional Tab will be shown called Optimization.

Optimization is always performed after you got a good GP model. After GP Modeling is finished, you have to set boundaries of input variables, set termination criteria (similar as in previous Run Tab), and check Minimum check box if you want to find minimum value of the model. Unchecked means you are finding Maximum value.

How To Set Min/Max value of Variables

1. Select variable form bottom grid in Optimize Model Tab, by clicking left mouse button.
2. In Min and Max text box input values

3. Press Update button.
4. Select another variable and perform previous steps.

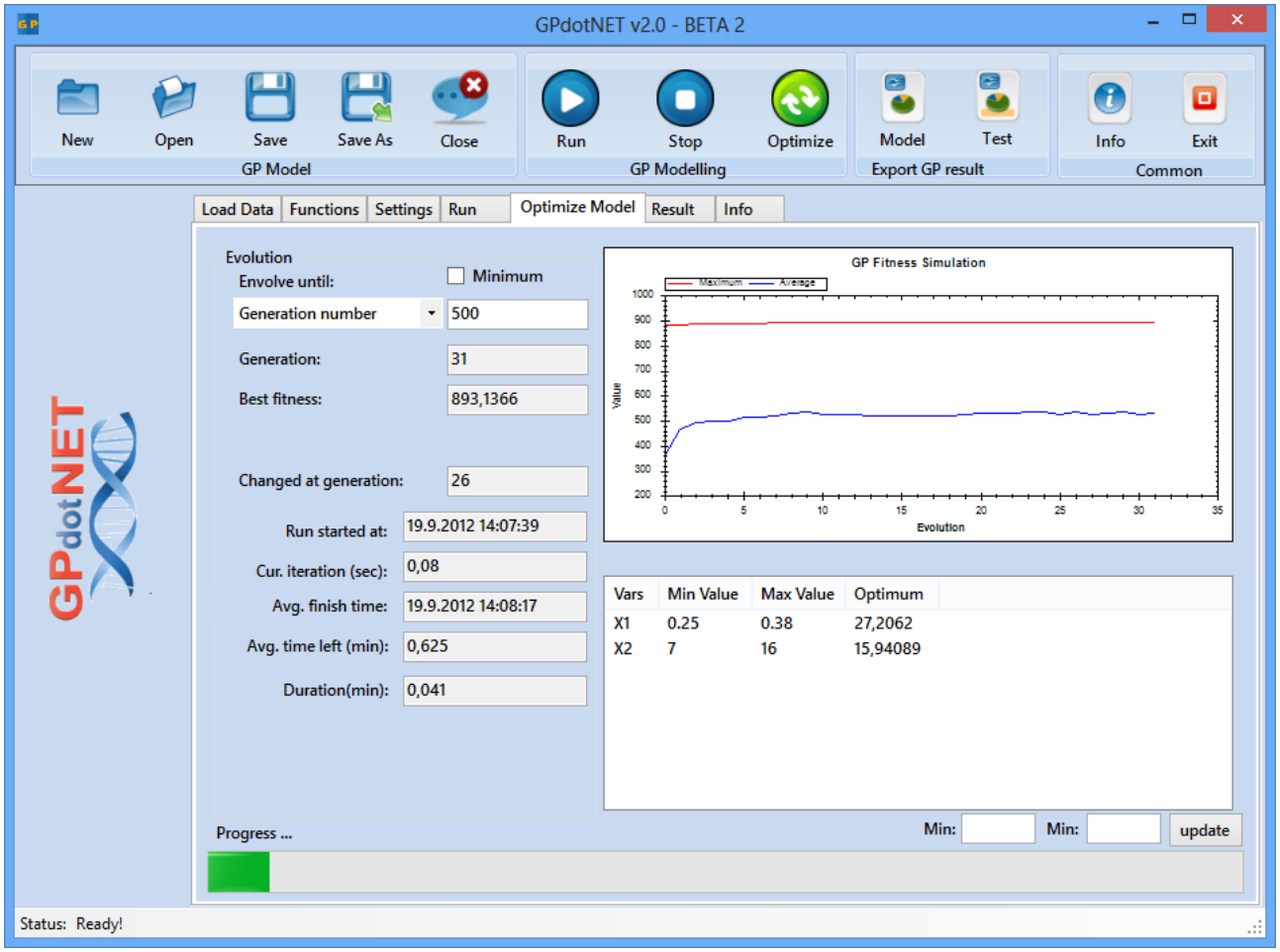


Figure 15: GPdotNET v2 Optimize Model Tab

6. GpdotNET Optimization of analitic function

The main new feature among several other is Optimization based on Genetic Algorithm. You can either optimize GPModel calculated in previously by using GP modeling, and also optimize analytically defined function defined with analytic function editor. The picture below shows sample of defining f(x)=x^3 – 6x^2 + 4x + 12 in the analytic function editor. During construction of the function, right table is filled automatically with variables and constants. End of process of defining analytic function is finished when the Finish button is pressed to transfer variables and constants in to Optimization panel.

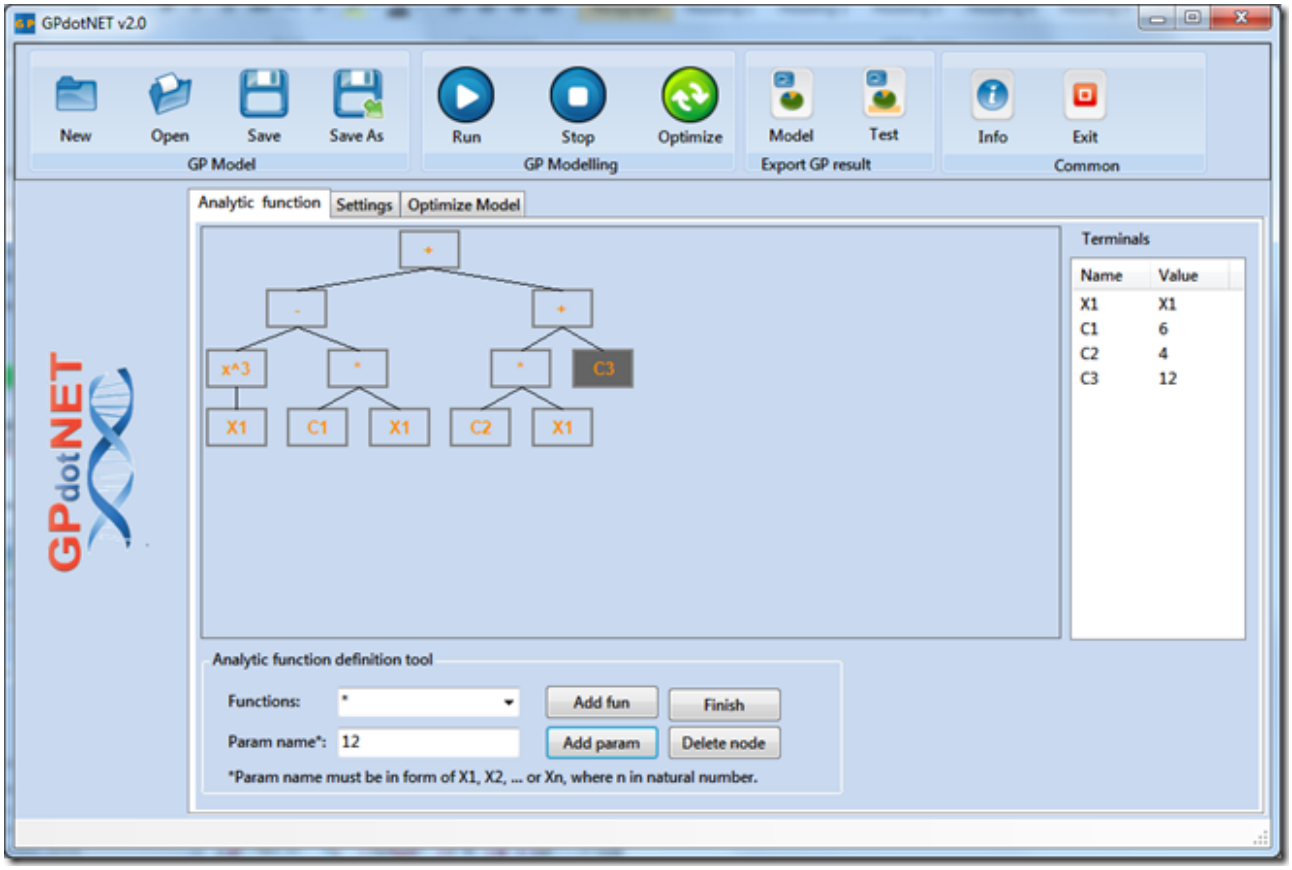


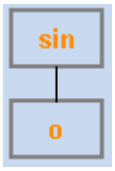
Figure 16: Analytic function editor in GPdotNET

After the Finish button is pressed, switch to Optimize model tab and define maximum and minimum values for input variables. The rest proces is the same as we have seen in previous chapter.

Working with Analytic function definition tool

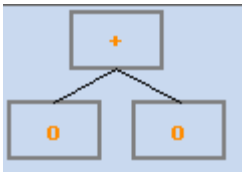
In addition with GPdotNET it will be possible to find global maximum or minimum of the function defined in analytic term as well. The picture below shows analytic tool definition. You can also see a brand new GUI for the GPdotNET v2.0.

Within GroupBox (see Fig. 12) you can see several buttons and combobox for selection basic math function. For example if you select SIN function from the combo box and click on Add function button, in the central window two rectangles will appear similar like this:



The picture represents sin function with one argument. O argument means that it is not defined yet, and you will not go further until you specified the name of the argument, or define a another function in chain. The tool automatically knows how many arguments need every defined function in GPdotNET.

If you select + or * function you will get three node, one with function name and two for arguments. Similar like this picture:



Defining function arguments

Every function must define its argument in order to works correctly. Letter small o in the second rectangle means that the argument of function sin is not defined yet. To define argument select node with left mouse click, enter the name of the argument, and click on Add Param button. Whole process is depicted on picture below.

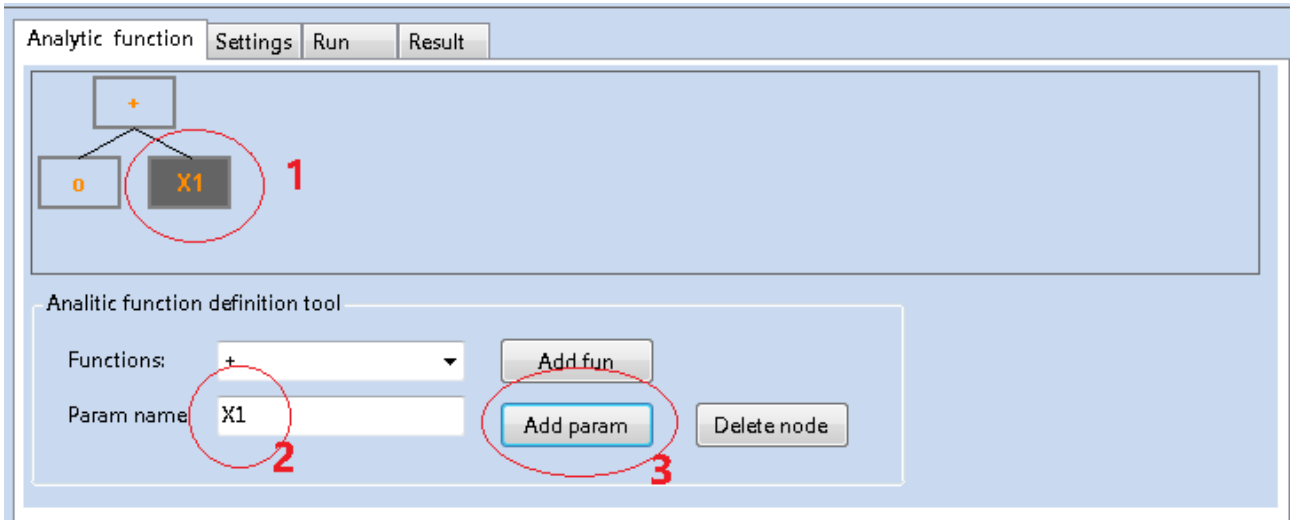


Figure 17: Adding function argument

Deleting the nodes in function

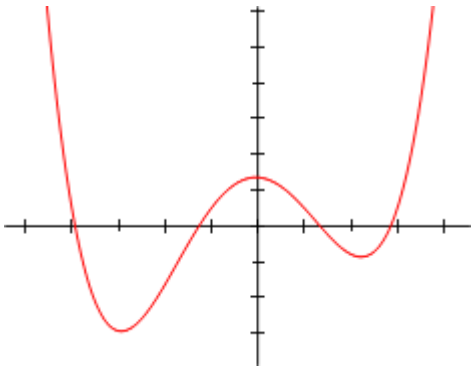
With Delete button you can delete nodes. In fact you can delete only leaf node. If you have bunch of nodes, and want to delete node in the middle, you need to delete all leaf node below it, in order to delete it. So select the leaf node with mouse and click on Delete node button.

After you finish the function definition, you must be sure that you defined all function nodes correctly with proper number of argument, otherwise GPdotNET will analise it and cannot pass further until you correct the function argument issue.

How to optimize analityc function in GpdotNET

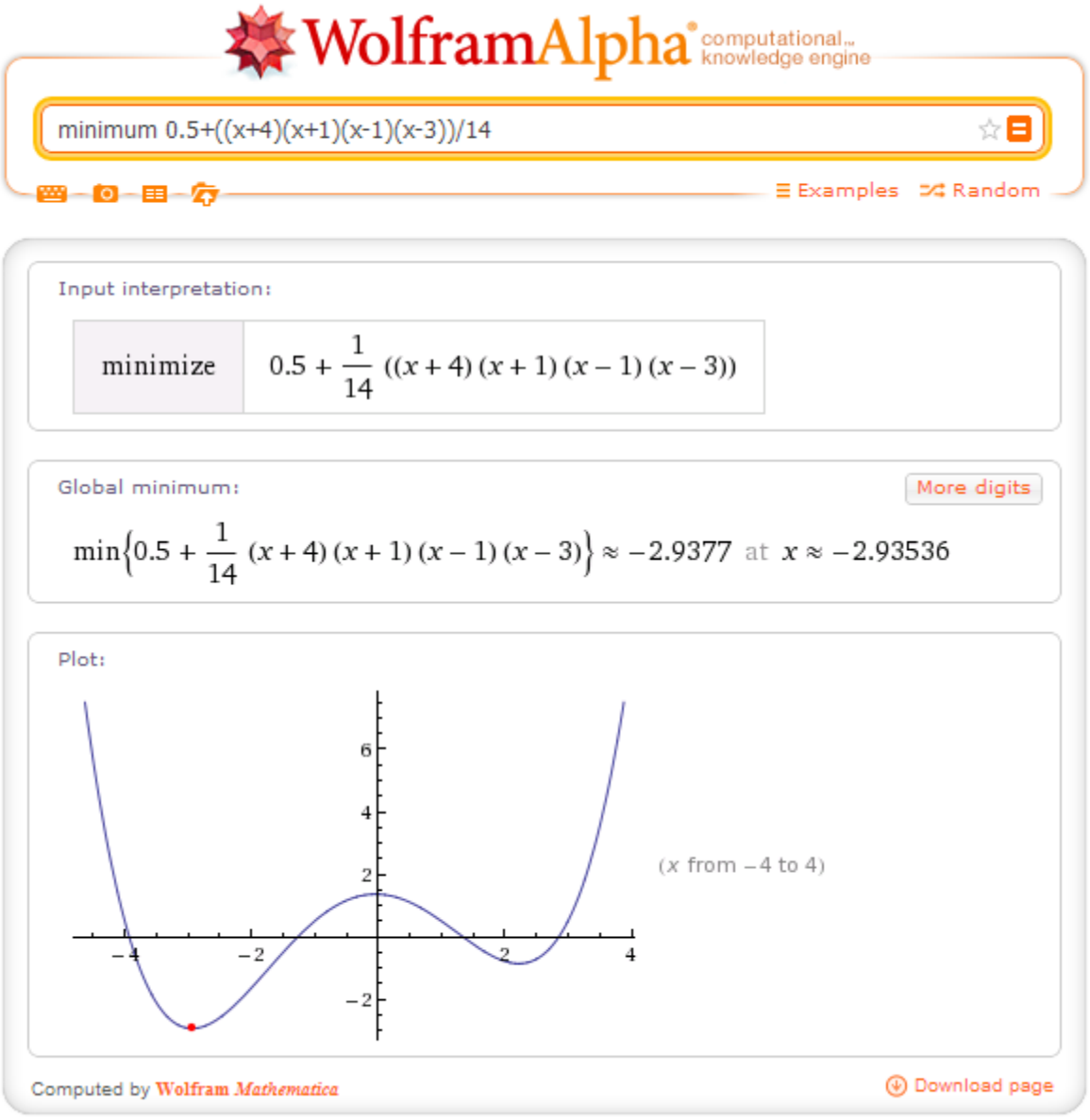
This is short tutorial how to use GPdotNET in order to find global optimum of analytically defined function.

Let's take an example of not so simple function. The picture below shows graph of the function. The function has two local minimum in interval from -5 to 5.



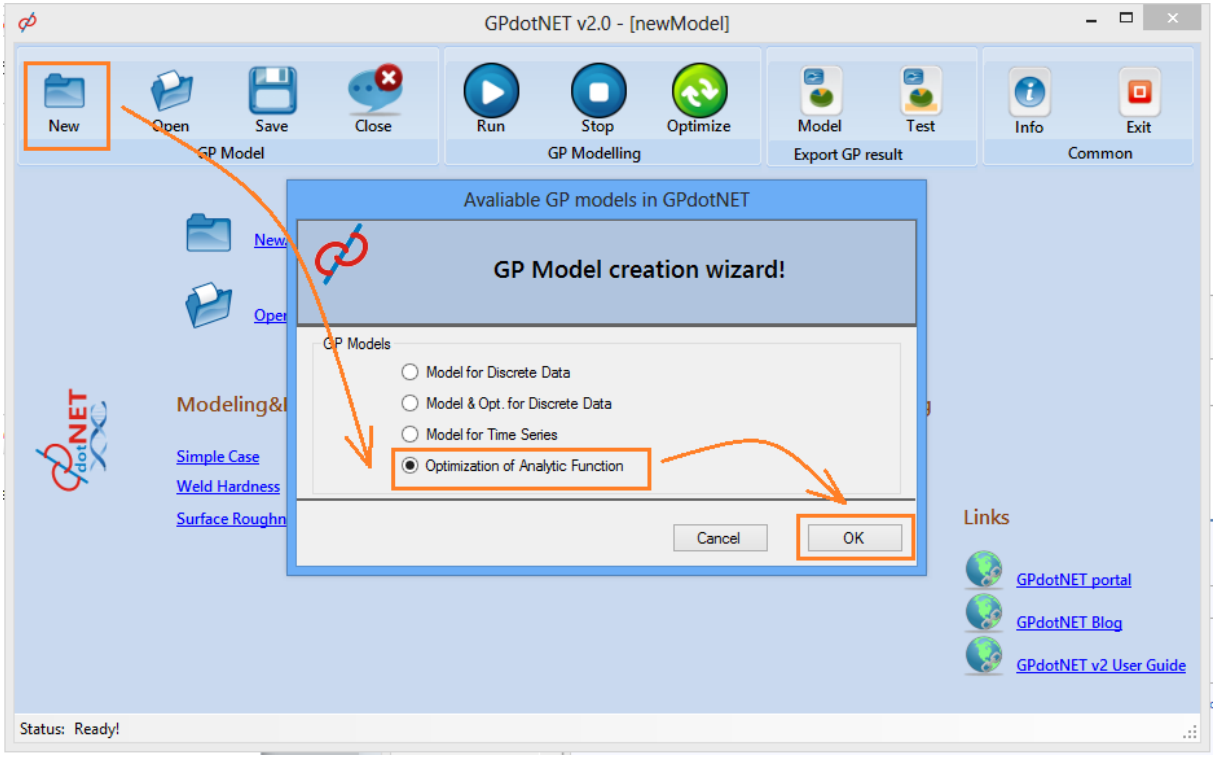
Analytic term of the function is: $y = \frac{(x+4)(x+1)(x-1)(x-3)}{14} + 0.5$.

To be sure that we dealing with correct result, first find optimum from Wolfram Alpha. The picture below shows global optimum of our function.

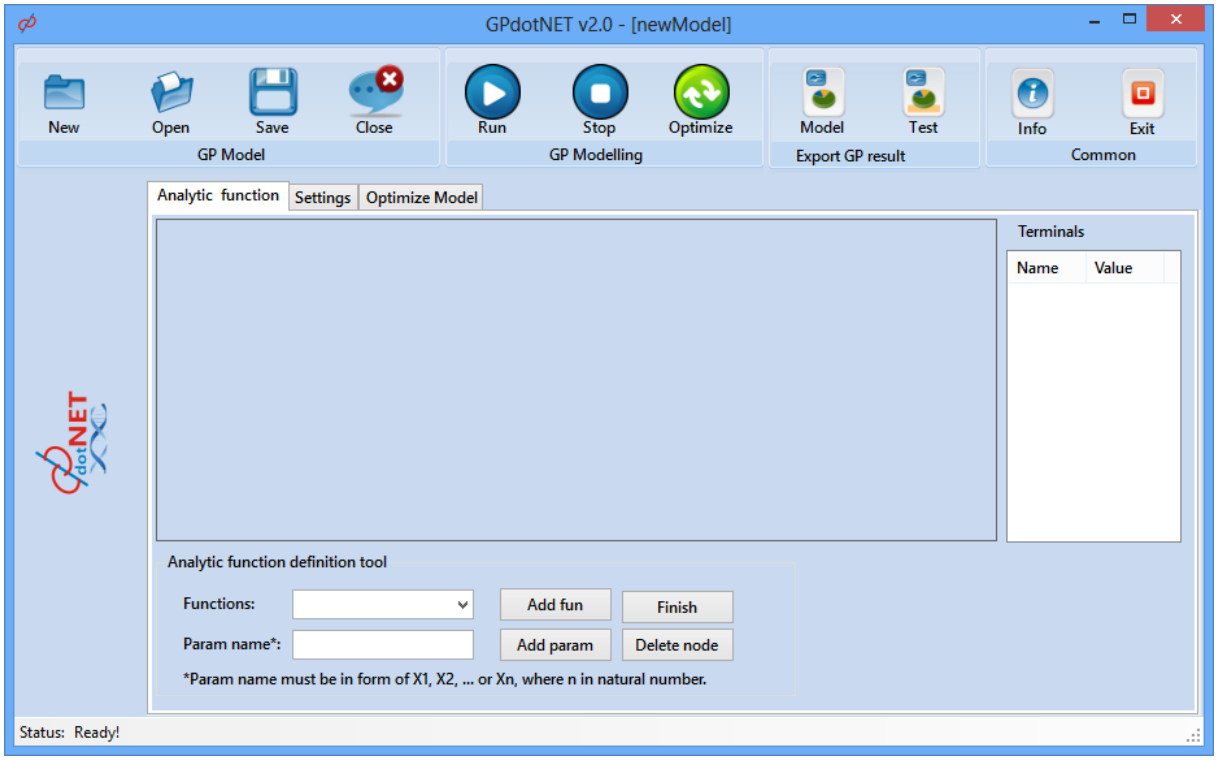


From the above we see that global minimum can be found for X=-2.93536, and corresponded optimal value is y=-2,377.

So let's try to find this optimum with GPdotNET v2.



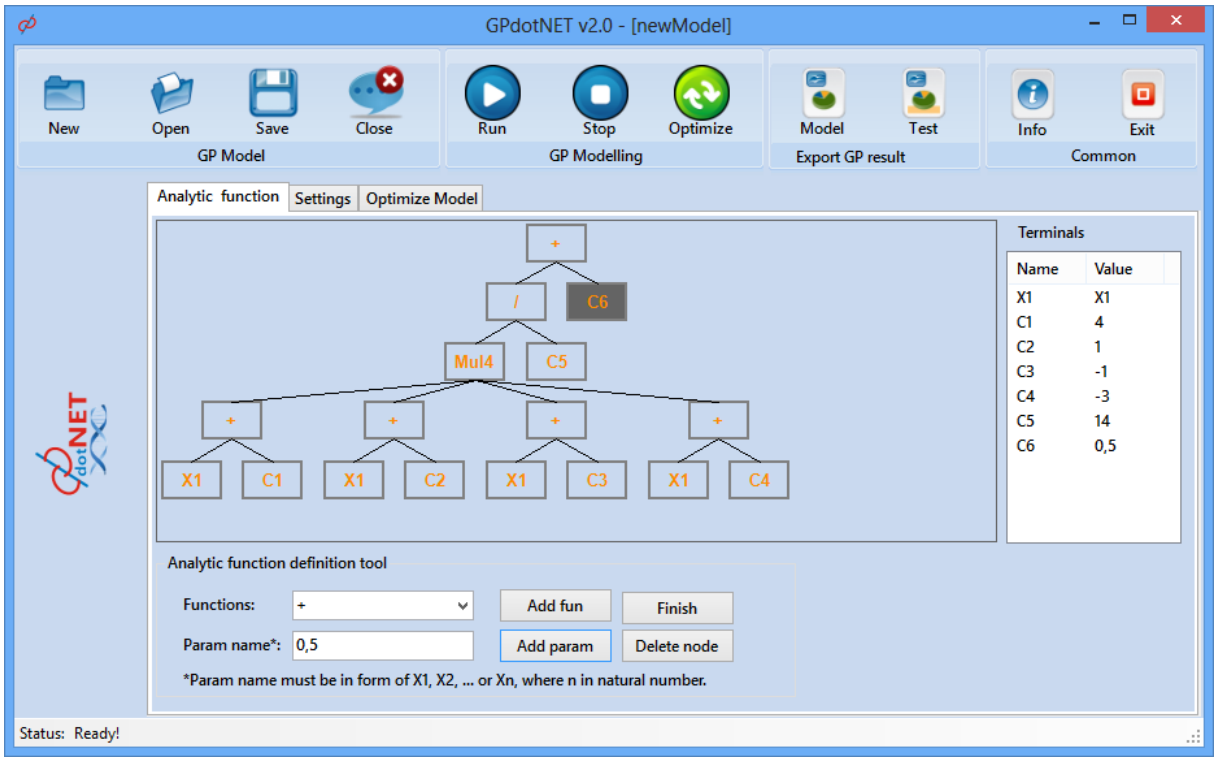
1. Open the GPdotNET and choose **New**.
2. Select **Optimization of Analytic Function** radio box
3. Click OK button.
4. Optimization module appears on the screen.



Now we need to define function in analytic form. As we have previously seen in blog post:

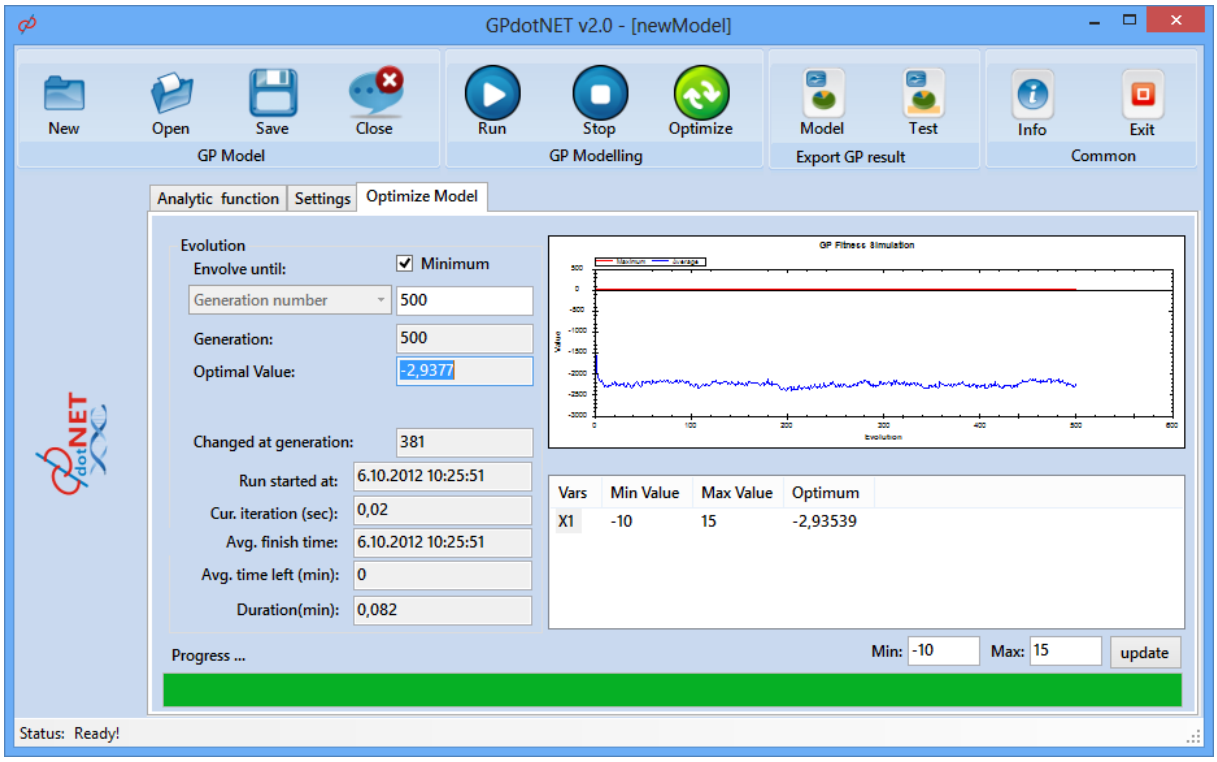
1. Select " +" function from Function combo box and press Add Function button.
2. Click on left outer node
3. Select "/" function and press Add Function button.
4. Select left outer node
5. Select **Mul4** function and press Add Function button.
6. For each outer node of Mult4 function add + function, similar like previous.
7. For each left outer node of + function , add Param Name **X1** and press **Add Param** buton.
8. For right node add 4,1,-1,-3 param from left to right.
9. For right node of Mul4 function add 14 param.
10. For right node of / function add 0,5 param.

After you finish you get the following picture:



11. Press Finish button and select Optimize Tab page.
12. Select X1 variable in List View control for defining variable range

13. Enter -10 for min and 15 for maximum and press **update** button.
14. Select Minimize check box, for minimum.
15. Now we can perform optimization by press **Optimize** toolbar icon.
16. After very short period of time you get the following picture which



From picture above we can see that we got Xmin=-2,935 and Ymin=-2,9377, exactly as we got when we have performed optimization by Wolfram Alpha.

7. GPdotNET File format

GPdotNET v2.0 introduce new textual file format for persisting GP and GA models and information during the program run. The file format is simple and easy, so sometimes you can edit in order to correct some minor changes. For example if you create GPModel without optimization, but later on you need to optimize this mode, you can easily change the type of model to enable optimization. The picture below shows sample GPdotNET "gpa" file format opened in Notepad++ with syntax highlighting.

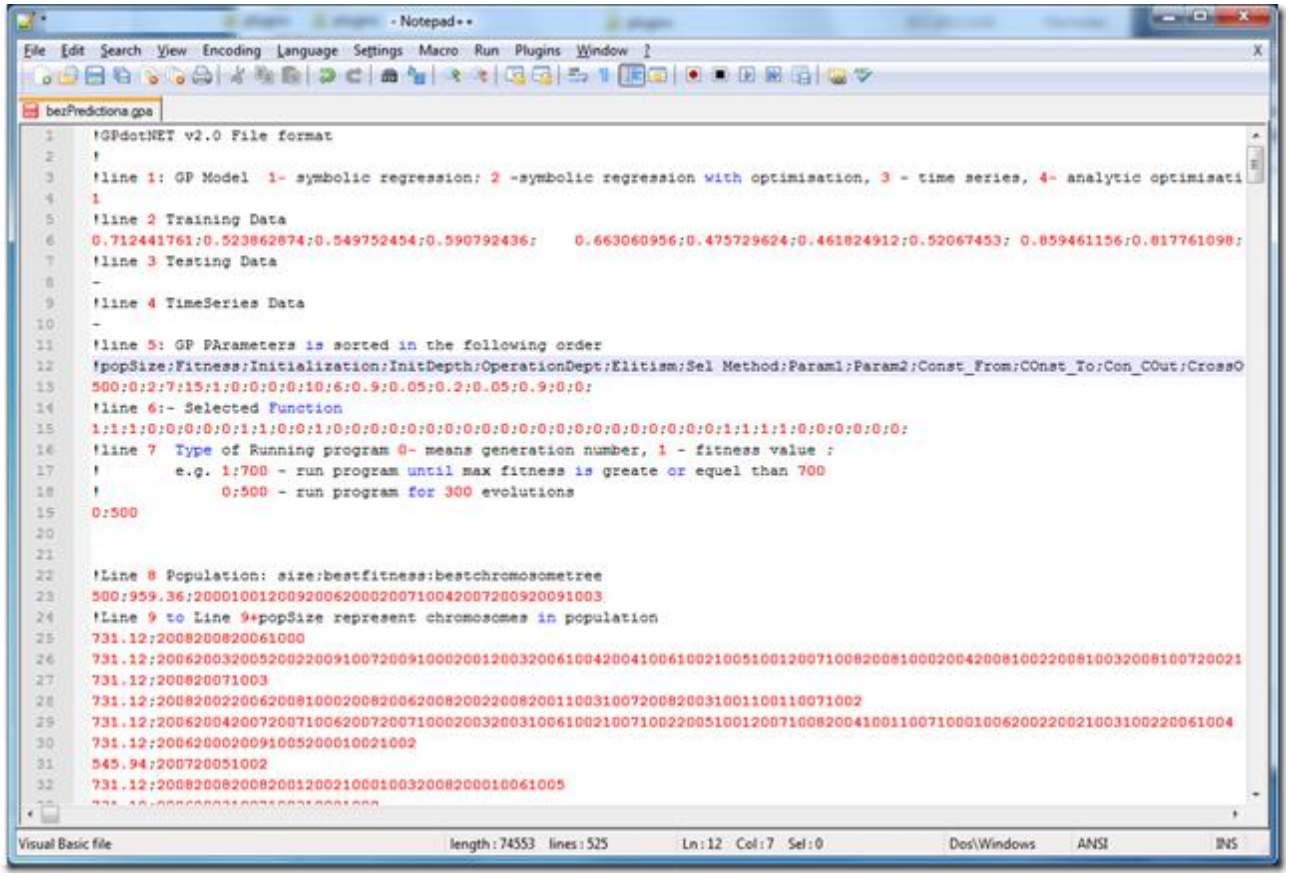


Figure 18: GPdotNET file format highlighted in Notepad ++

8. Exporting Results in GpdotNET

When the model is calculated, you ca export training and testing dana in to Excel (only for Windows user) and csv file format.

When you export to Excel you can export GPModel in form of Excel formula, for further analysis.

Choose model icon if you want fo export training data and GPModel. Otherwise choose Test icon for exporting testing data.

After you choose right export icon, export dialog appears:

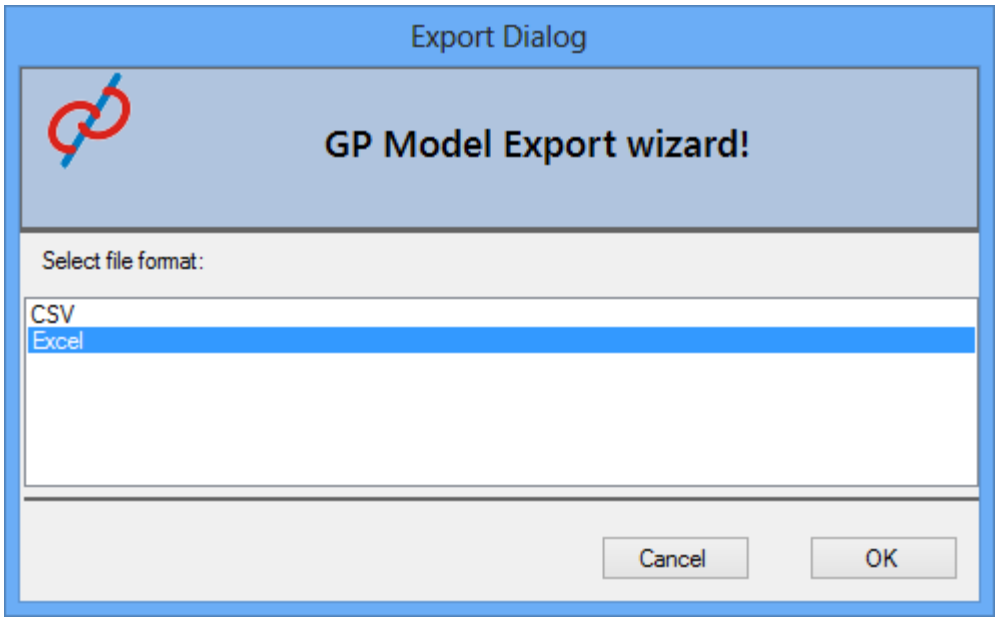
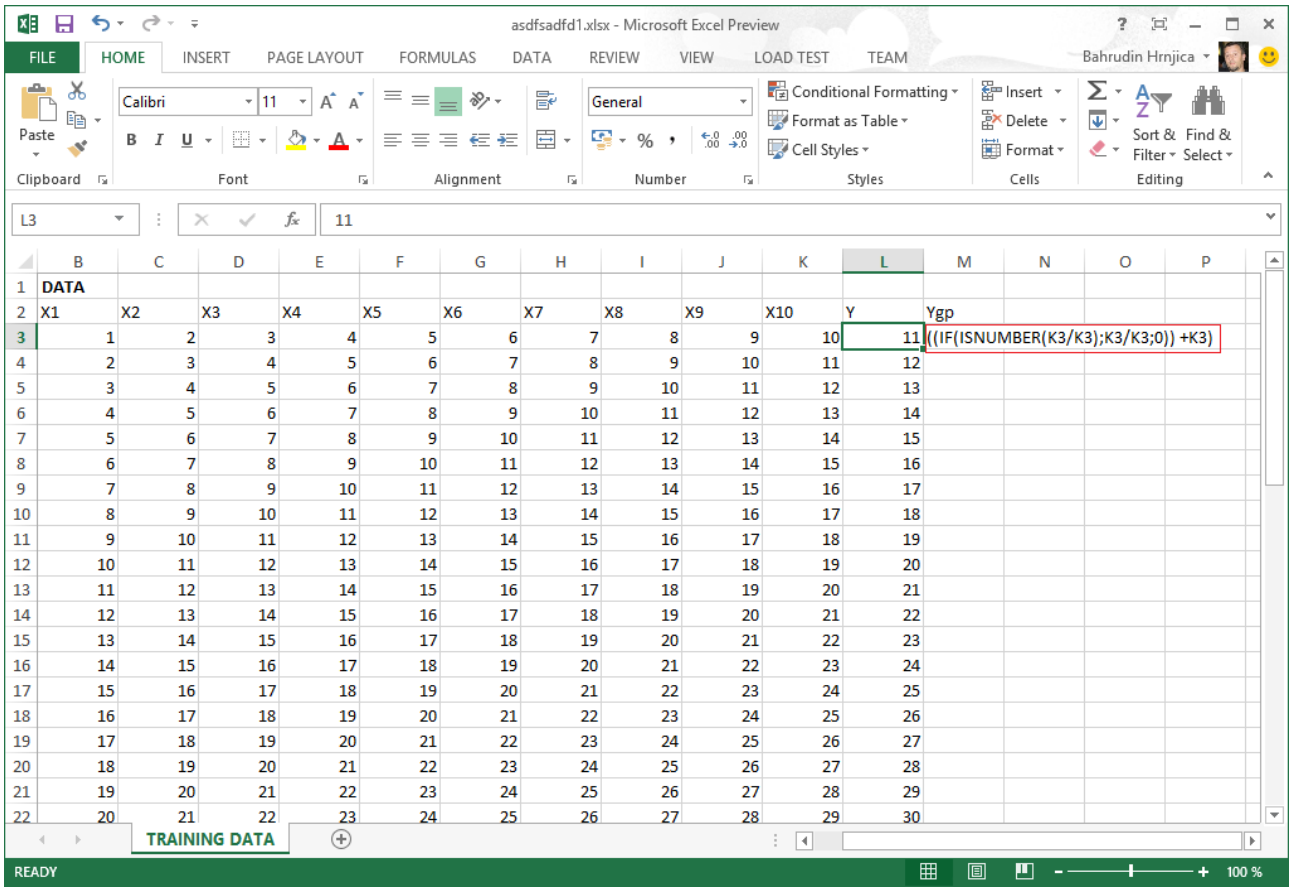


Figure 19: Export dialog

After you select right export format, Save dialog appears, to choosing right file nname and path.

In case of Excel exporting GP Model column is as text showing formula. You need to put equal sigh in front of content in order that model be calculated. The picture below show similar case.

The reason why you need to put = sign is that sometime formula is too long, and cannot be pasted in to cell.



	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	DATA														
2	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	Y	Ygp			
3	1	2	3	4	5	6	7	8	9	10	11	((IF(ISNUMBER(K3/K3);K3/K3;0)) +K3)			
4	2	3	4	5	6	7	8	9	10	11	12				
5	3	4	5	6	7	8	9	10	11	12	13				
6	4	5	6	7	8	9	10	11	12	13	14				
7	5	6	7	8	9	10	11	12	13	14	15				
8	6	7	8	9	10	11	12	13	14	15	16				
9	7	8	9	10	11	12	13	14	15	16	17				
10	8	9	10	11	12	13	14	15	16	17	18				
11	9	10	11	12	13	14	15	16	17	18	19				
12	10	11	12	13	14	15	16	17	18	19	20				
13	11	12	13	14	15	16	17	18	19	20	21				
14	12	13	14	15	16	17	18	19	20	21	22				
15	13	14	15	16	17	18	19	20	21	22	23				
16	14	15	16	17	18	19	20	21	22	23	24				
17	15	16	17	18	19	20	21	22	23	24	25				
18	16	17	18	19	20	21	22	23	24	25	26				
19	17	18	19	20	21	22	23	24	25	26	27				
20	18	19	20	21	22	23	24	25	26	27	28				
21	19	20	21	22	23	24	25	26	27	28	29				
22	20	21	22	23	24	25	26	27	28	29	30				

Figure 20: Excel showing GPModel as excel formul. You need to type = in order to get right formula.