GPdotNET v2.0 User Guide

bhrnjica
Nov, 2012

Software Disclaimer

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Project description

GPdotNET is artificial intelligence tool for applying Genetic Programming and Genetic Algorithm in modeling and optimization of various engineering problems. It is .NET (Mono) application written in C# programming language which can run on both Windows and Linux based OS. Project started in 2006 within postgraduate project for modeling and optimization with evolutionary algorithms. As open source project, GPdotNET is first published on November 5 2009 on codeplex.com. GPdotNET is very easy to use. Even if you have no deep knowledge of GP and GA, you can apply those methods in finding solution. The project can be used 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. 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 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.

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. Modeling with or without prediction of discrete data by using Symbolic Regression modeling with GP
2. Modeling with or without prediction of discrete data by using Symbolic Regression with GP and Optimizing calculated GPdotNET model by using GA
3. Time Series modeling and prediction data by using Symbolic Regression with GP
4. Optimization of analytical defined function by using GA

Introduction to GPdotNET v2

During implementation of GPdotNET v2.0 I have tried to post as much information as I could, trying to provide all relevant information to the users. This user guide mostly contains my blog posts, I was writing during implementations. Comparing to the previous version, this version is completely rewritten, and it is not compatible with version 1.0 in any segment. General logic of applying evolutionary methods in solving engineering problem remain the same. In the following text, it is listed the main feature in GPdotNEt v2.0.

Cross OS and Cross platform software (new in v2)

One of the main requirement for GPdotNET v2.0 is ability to run on multiple OS, 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. When code is not compatible with Mono, it was replaced with the code available in Mono. I can say that the whole implementation is 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 1, so every bug report would be appreciated.

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 is some compatibility issue in Mono, so you cannot use Excel exporting in Mono. While you ruinning 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 modelling 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 Modelling at the same time.

Optimization of analytically defined function (new in v2)

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

Info tab in Model (new in v2)

When you start with modelling 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, by removing all unnecessary options. Commands are split in to 4 major groups: Model, Modelling, 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 use ability to stop or run programs from any tab page, not only from run page.

Working with GPdotNET v2

Picture below show 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.

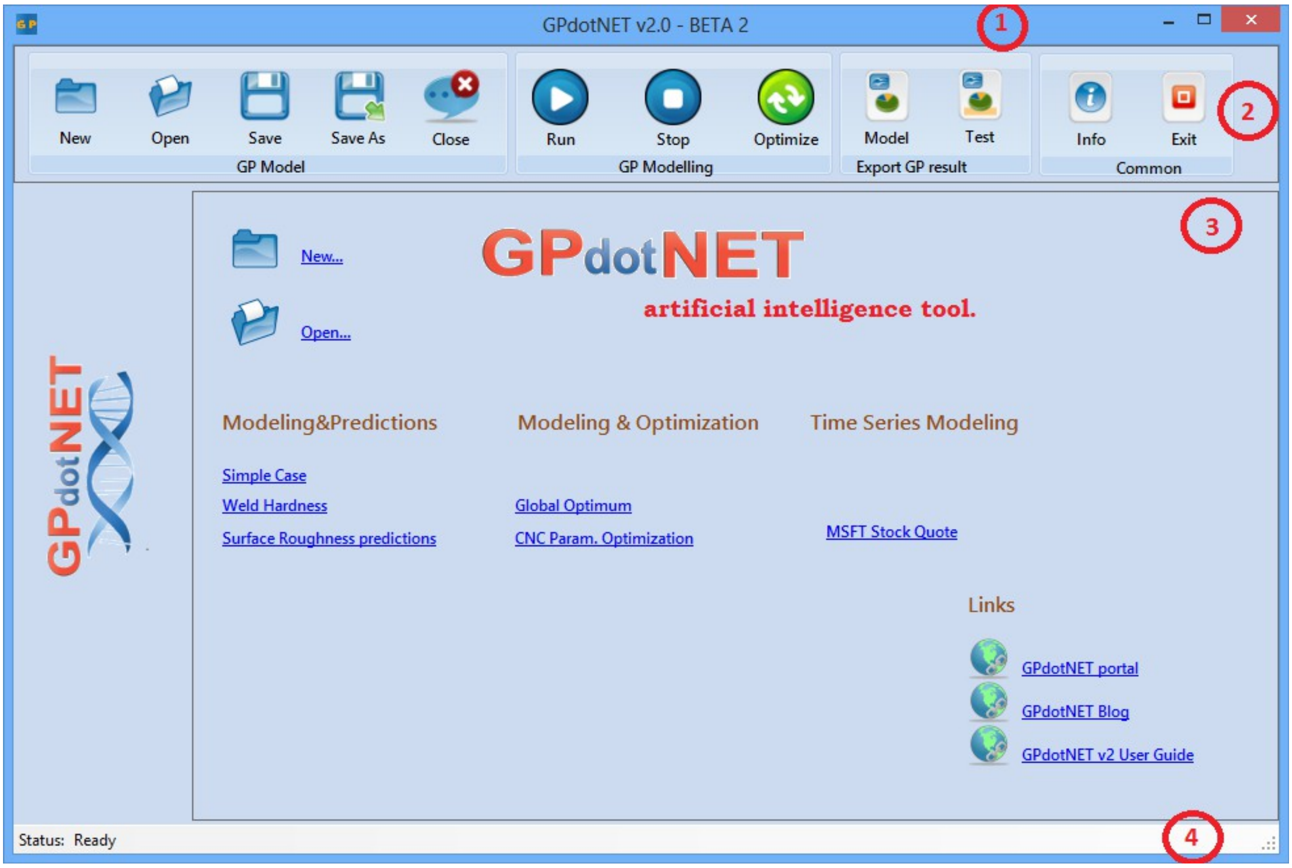


Figure 1: GPdotNET Start Screen

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. GPModel – gather 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.
2. 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.
3. Export GP Result contains options for exporting result to other format: Excel and CSV.
4. 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. 1).

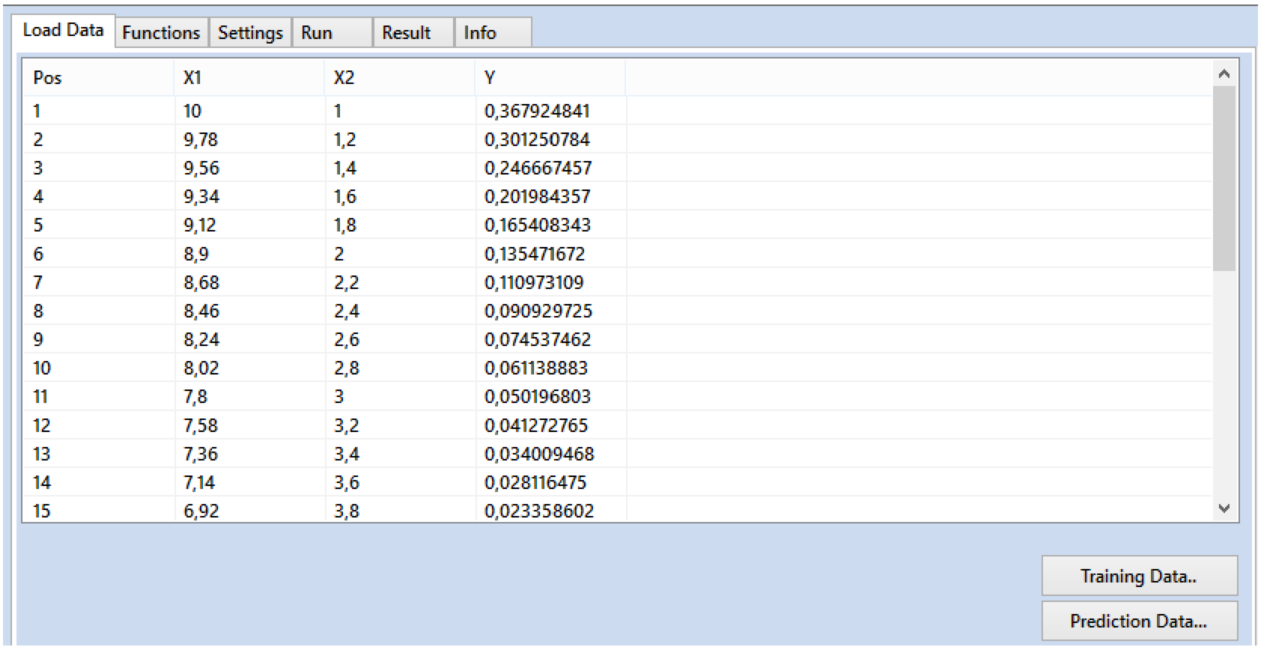
1. Modeling & predictions
2. Modeling & Optimization
3. Time Series Modeling

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

The last group of links are 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.



Pos	X1	X2	Y
1	10	1	0,367924841
2	9,78	1,2	0,301250784
3	9,56	1,4	0,246667457
4	9,34	1,6	0,201984357
5	9,12	1,8	0,165408343
6	8,9	2	0,135471672
7	8,68	2,2	0,110973109
8	8,46	2,4	0,090929725
9	8,24	2,6	0,074537462
10	8,02	2,8	0,061138883
11	7,8	3	0,050196803
12	7,58	3,2	0,041272765
13	7,36	3,4	0,034009468
14	7,14	3,6	0,028116475
15	6,92	3,8	0,023358602

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.

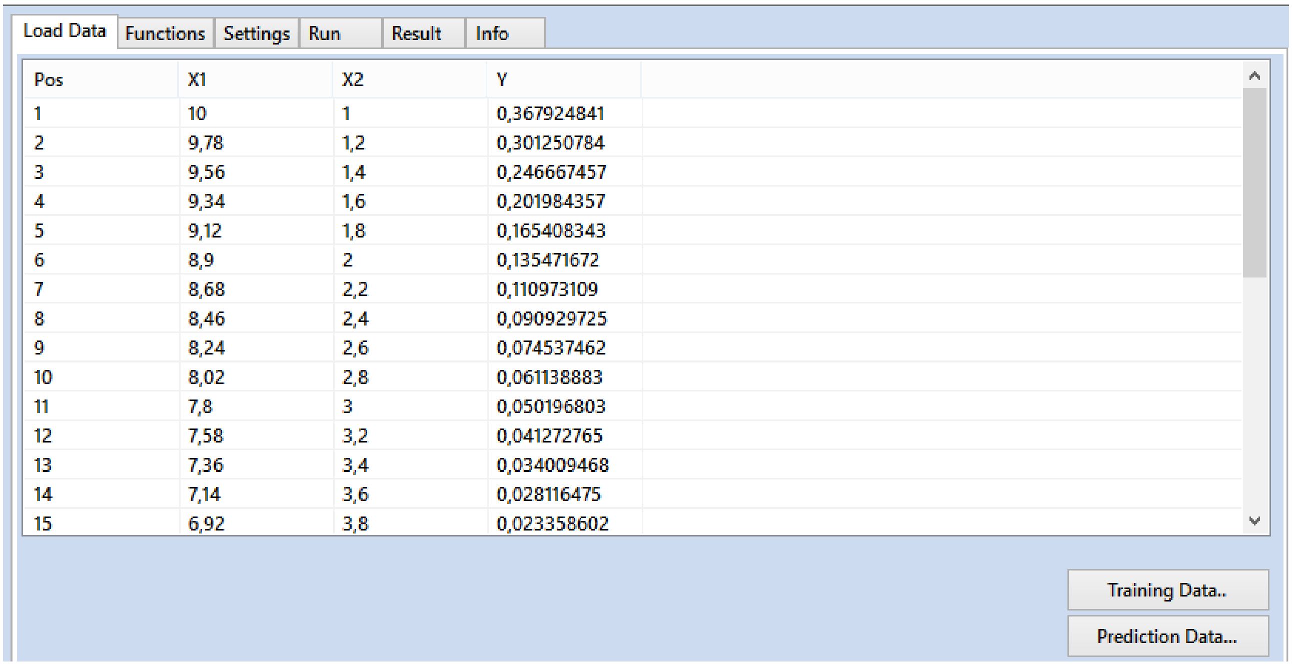
Load Data Tab

The Load Data tab control (see Fig. 4) appears in every type of modeling. This is one of the main tab. 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.

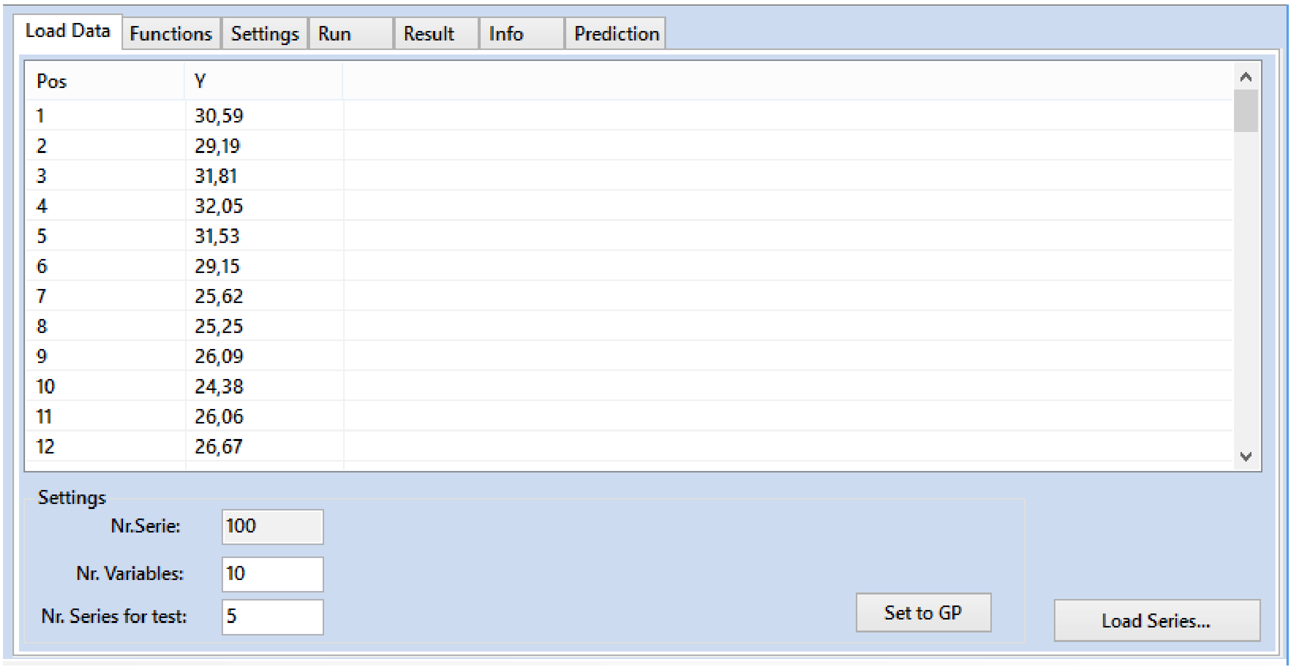


Pos	X1	X2	Y
1	10	1	0,367924841
2	9,78	1,2	0,301250784
3	9,56	1,4	0,246667457
4	9,34	1,6	0,201984357
5	9,12	1,8	0,165408343
6	8,9	2	0,135471672
7	8,68	2,2	0,110973109
8	8,46	2,4	0,090929725
9	8,24	2,6	0,074537462
10	8,02	2,8	0,061138883
11	7,8	3	0,050196803
12	7,58	3,2	0,041272765
13	7,36	3,4	0,034009468
14	7,14	3,6	0,028116475
15	6,92	3,8	0,023358602

Figure 2: 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.)



Pos	Y
1	30,59
2	29,19
3	31,81
4	32,05
5	31,53
6	29,15
7	25,62
8	25,25
9	26,09
10	24,38
11	26,06
12	26,67

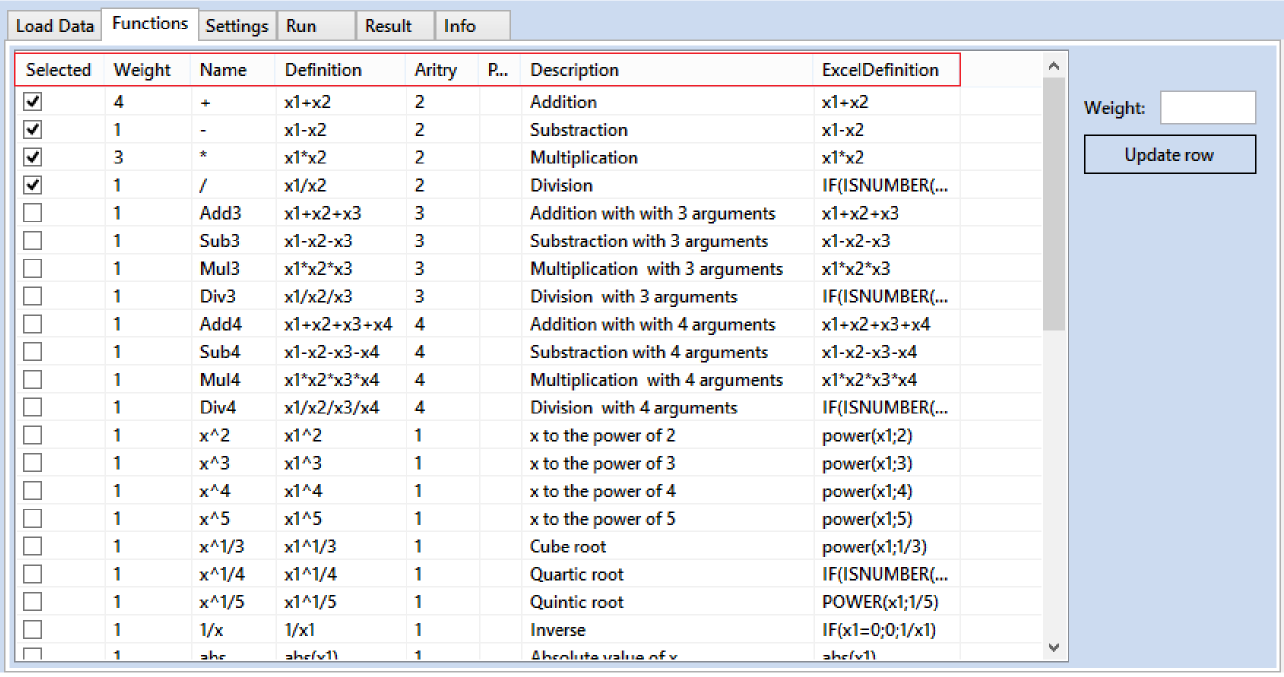
Nr.Serie: 100
Nr. Variables: 10
Nr. Series for test: 5

Figure 3: 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 4: 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.

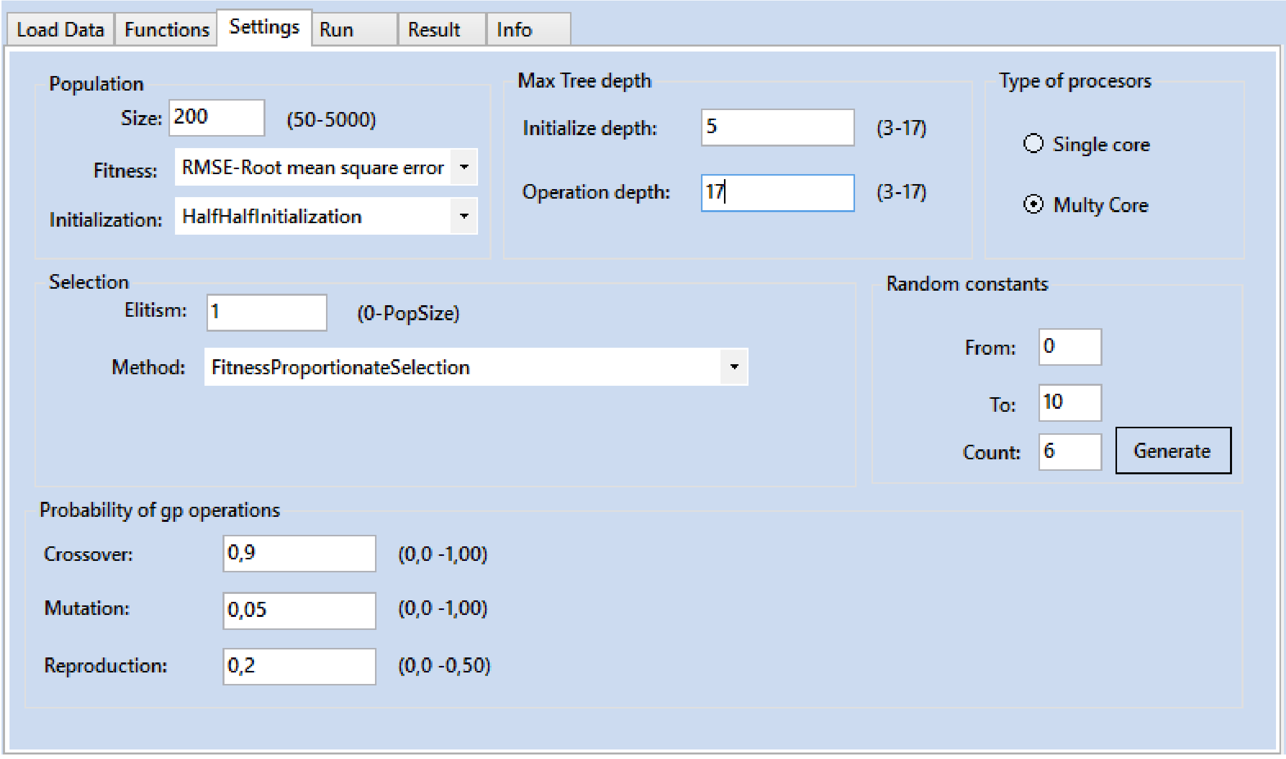
Changing Weight of the function

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.



Population

Size: 200 (50-5000)

Fitness: RMSE-Root mean square error

Initialization: HalfHalfInitialization

Max Tree depth

Initialize depth: 5 (3-17)

Operation depth: 17 (3-17)

Type of procesors

○ Single core

● Multy Core

Selection

Elitism: 1 (0-PopSize)

Method: FitnessProportionateSelection

Random constants

From: 0

To: 10

Count: 6

Probability of gp operations

Crossover: 0,9 (0,0 -1,00)

Mutation: 0,05 (0,0 -1,00)

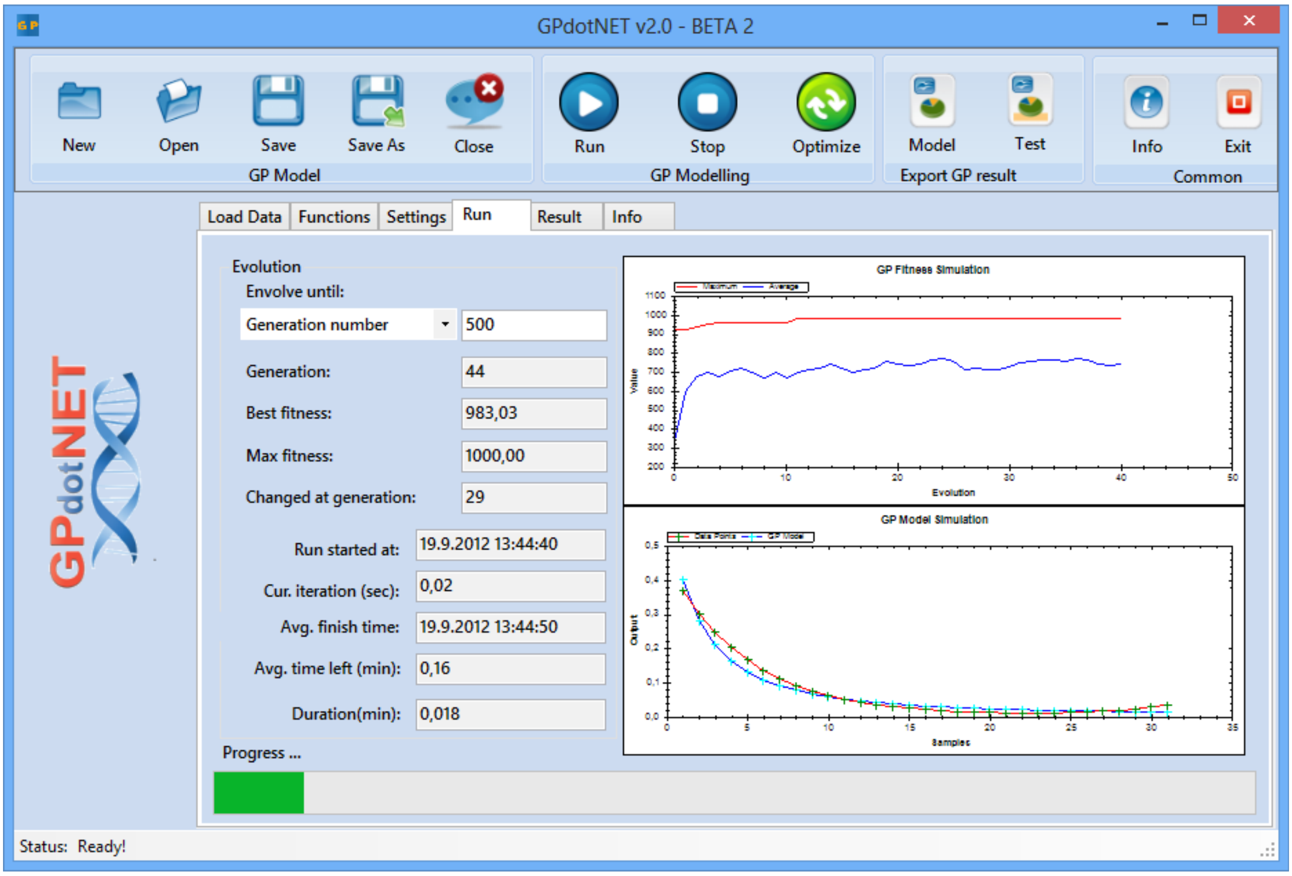
Reproduction: 0,2 (0,0 -0,50)

Figure 5: 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.



Evolution

Envolve until:

Generation number 500

Generation: 44

Best fitness: 983,03

Max fitness: 1000,00

Changed at generation: 29

Run started at: 19.9.2012 13:44:40

Cur. iteration (sec): 0,02

Avg. finish time: 19.9.2012 13:44:50

Avg. time left (min): 0,16

Duration(min): 0,018

Status: Ready!

Figure 6: 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

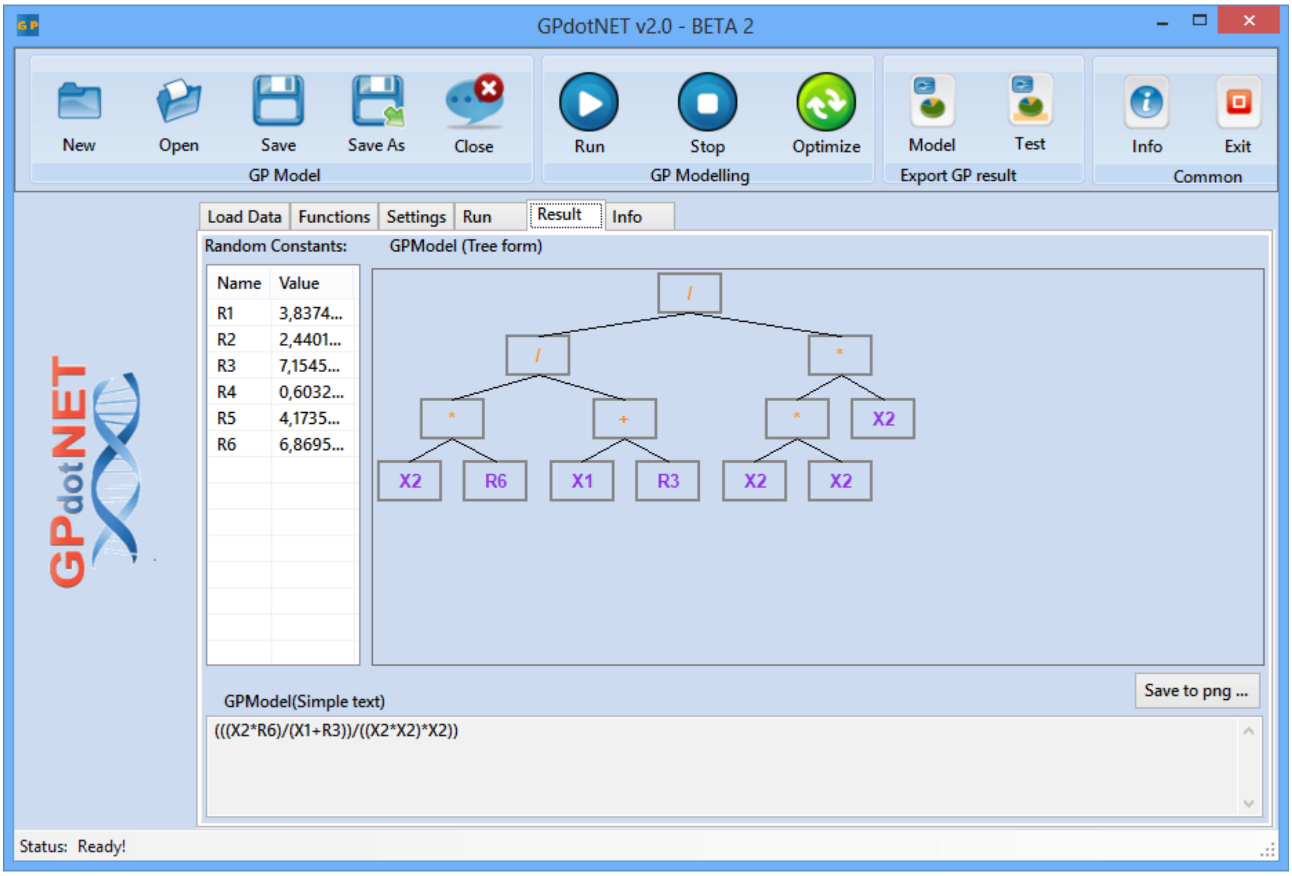
Setting Termination Criteria

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.



Random Constants:

Name	Value
R1	3,8374...
R2	2,4401...
R3	7,1545...
R4	0,6032...
R5	4,1735...
R6	6,8695...

GPModel(Simple text)

```
(((X2*R6)/(X1+R3))/((X2*X2)*X2))
```

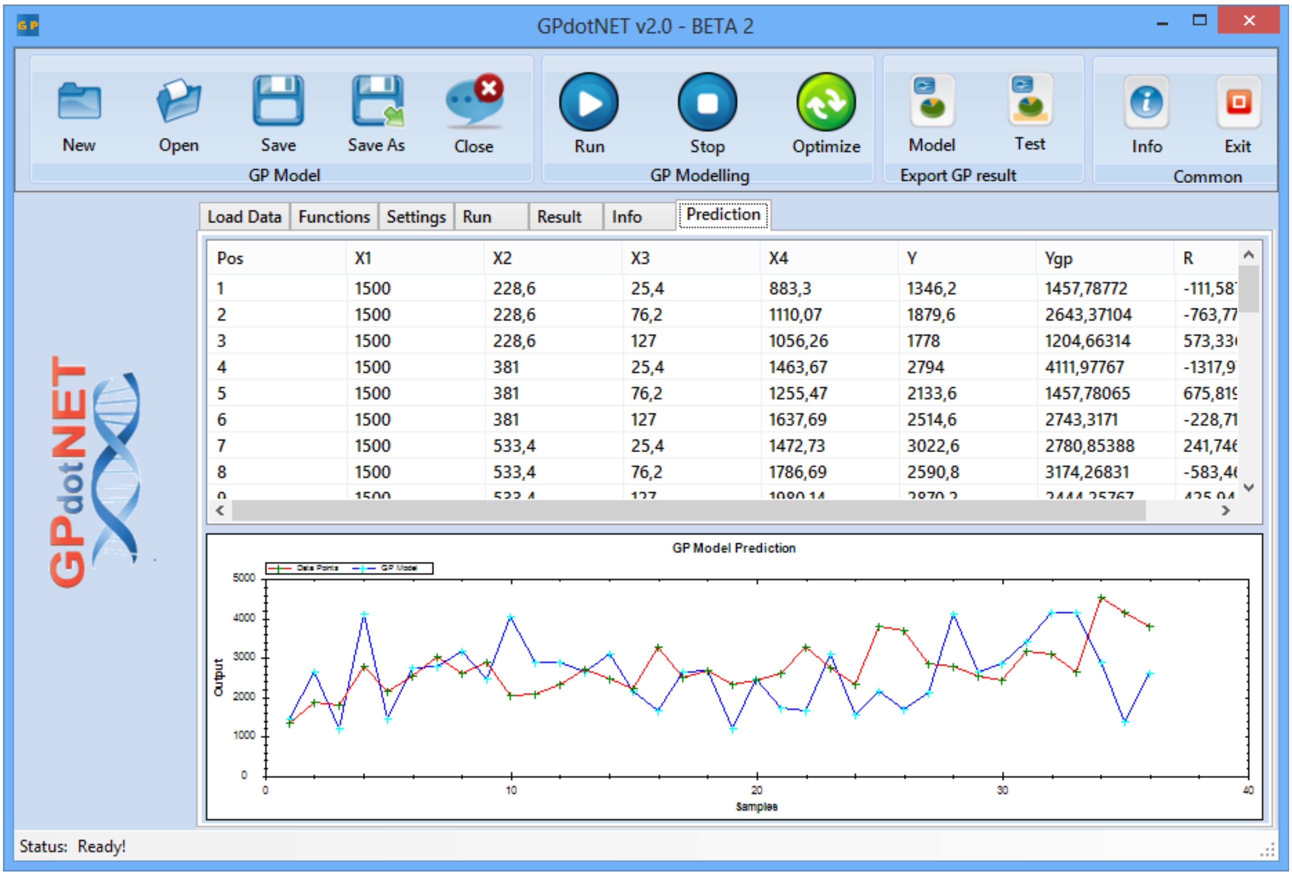
Status: Ready!

Figure 7: 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 8 shows Prediction Tab, which contains Table and Chart of predicted data calculated with the current solution.

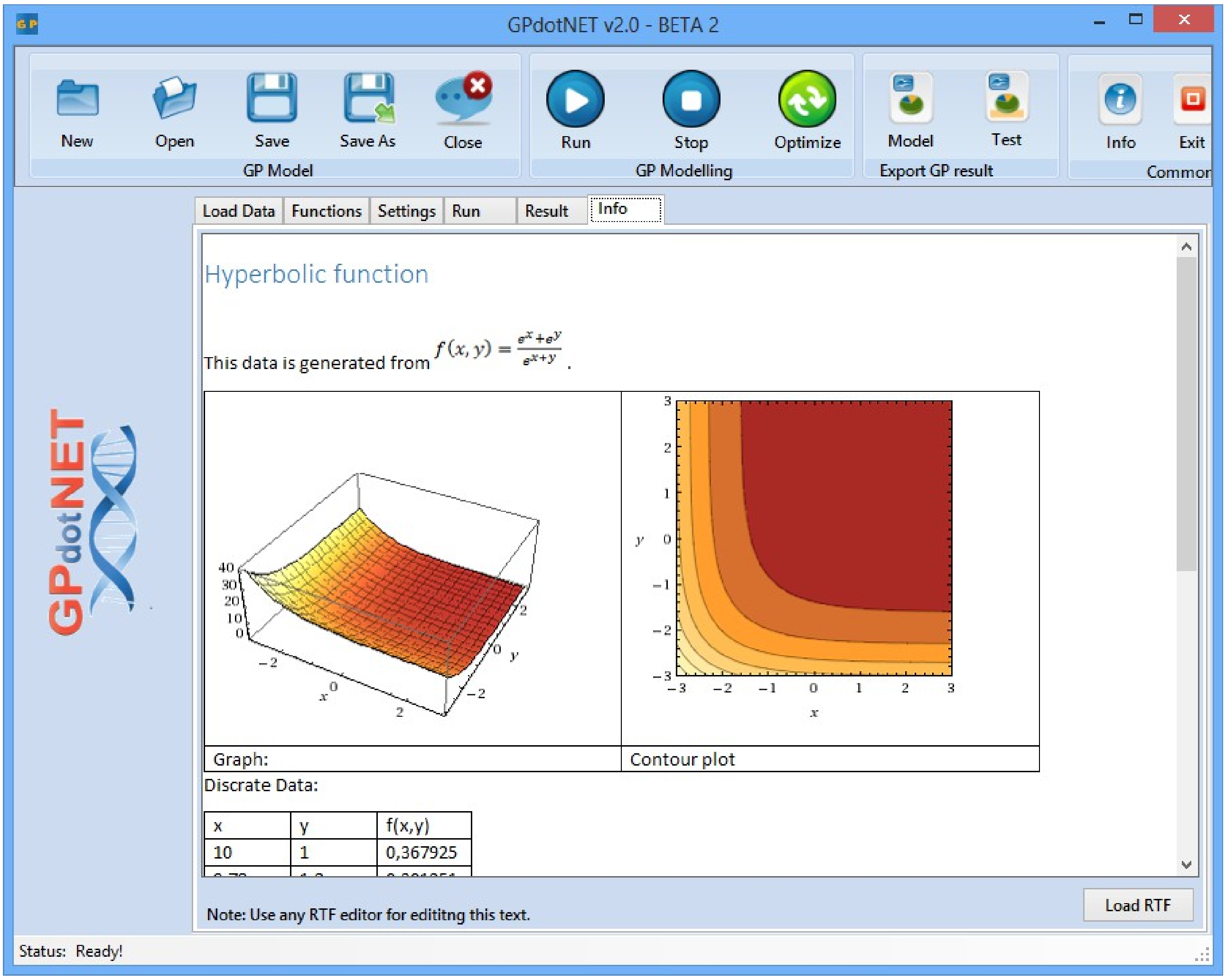


Pos	X1	X2	X3	X4	Y	Ygp	R
1	1500	228,6	25,4	883,3	1346,2	1457,78772	-111,58
2	1500	228,6	76,2	1110,07	1879,6	2643,37104	-763,77
3	1500	228,6	127	1056,26	1778	1204,66314	573,33
4	1500	381	25,4	1463,67	2794	4111,97767	-1317,9
5	1500	381	76,2	1255,47	2133,6	1457,78065	675,81
6	1500	381	127	1637,69	2514,6	2743,3171	-228,71
7	1500	533,4	25,4	1472,73	3022,6	2780,85388	241,74
8	1500	533,4	76,2	1786,69	2590,8	3174,26831	-583,4
9	1500	533,4	127	1980,14	2870,2	2444,25767	425,94

Figure 8: 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.



Hyperbolic function

This data is generated from $f(x,y) = \frac{e^x + e^y}{e^{x+y}}$.

Graph:

Contour plot

Discrate Data:

x	y	f(x,y)
10	1	0,367925

Note: Use any RTF editor for edititng this text.

Load RTF

Status: Ready!

Figure 9: GPdotNET v2 Info Tab

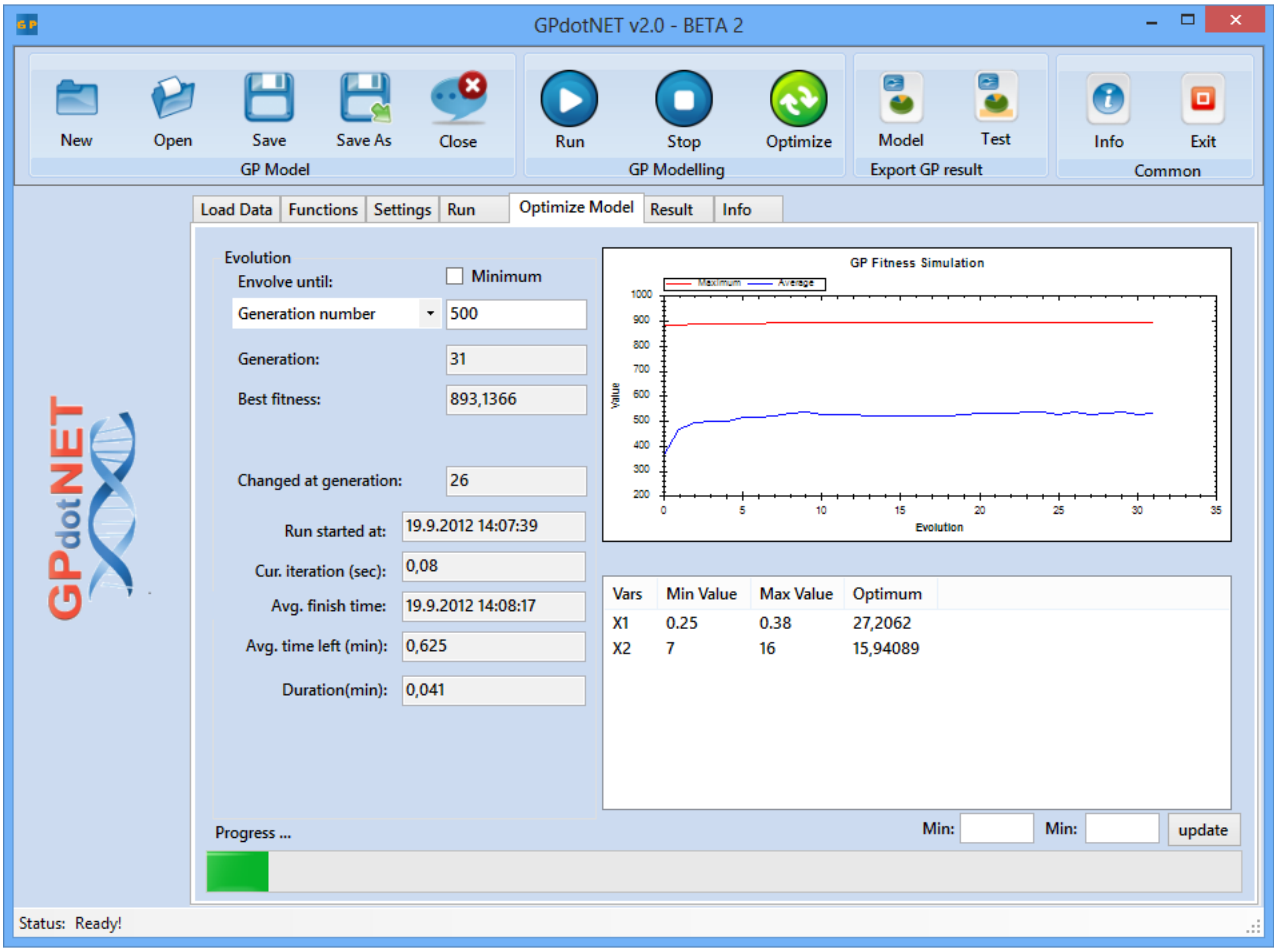
Optimize Model Tab

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.



Evolution

Envolve until: □ Minimum

Generation number: 500

Generation: 31

Best fitness: 893,1366

Changed at generation: 26

Run started at: 19.9.2012 14:07:39

Cur. iteration (sec): 0,08

Avg. finish time: 19.9.2012 14:08:17

Avg. time left (min): 0,625

Duration(min): 0,041

Vars	Min Value	Max Value	Optimum
X1	0.25	0.38	27,2062
X2	7	16	15,94089

Min:

Min:

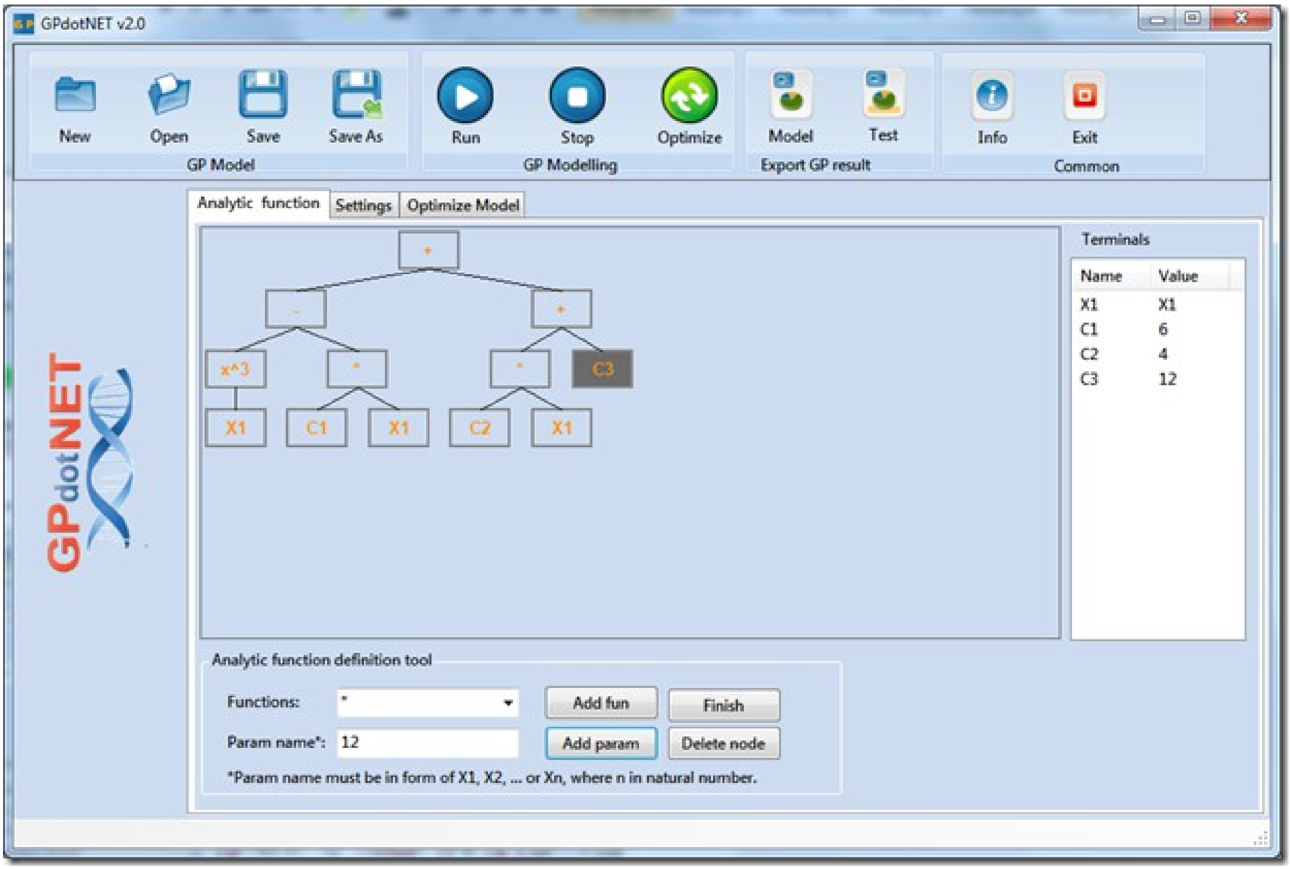
Progress ...

Status: Ready!

Figure 10: GPdotNET v2 Optimize Model Tab

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.



Name	Value
X1	X1
C1	6
C2	4
C3	12

Functions: *

Param name*: 12

*Param name must be in form of X1, X2, ... or Xn, where n in natural number.

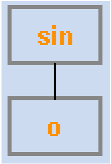
Figure 11: Analytic function editor in GPdotNET

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

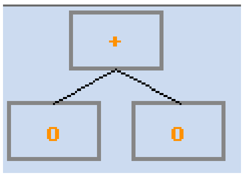
In addition with GPdotNET it will be possible to find global maximum or minimum of the function defined in analytic term as well. The picture below shows analytic tool definition. You can also see a brand new GUI for the GPdotNET v2.0.

Within GroupBox (see Fig. 11) 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.

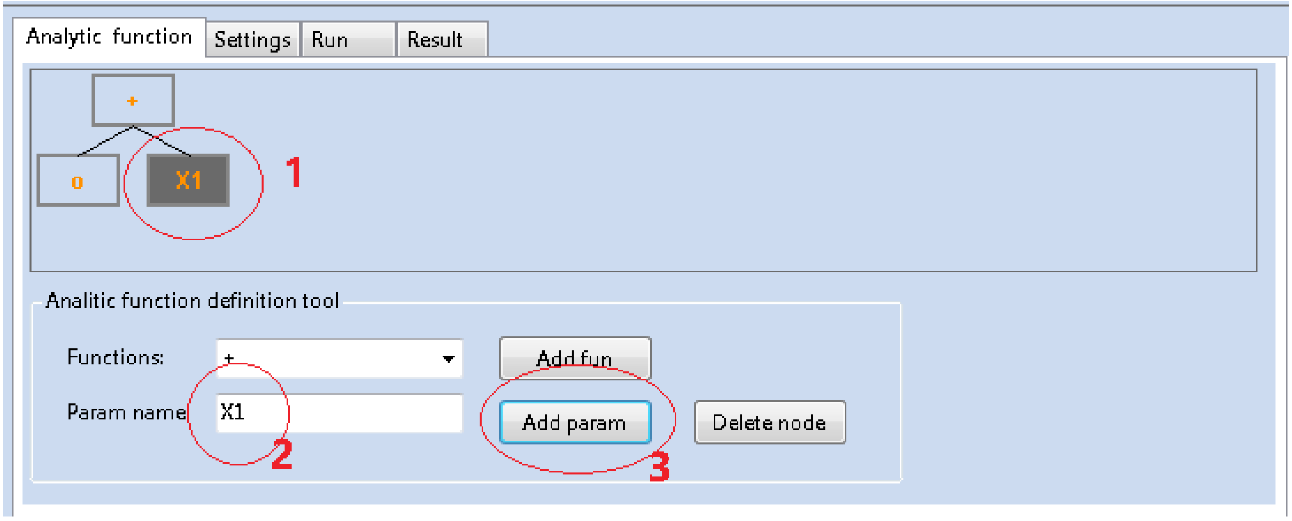


Figure 12: Adding function argument

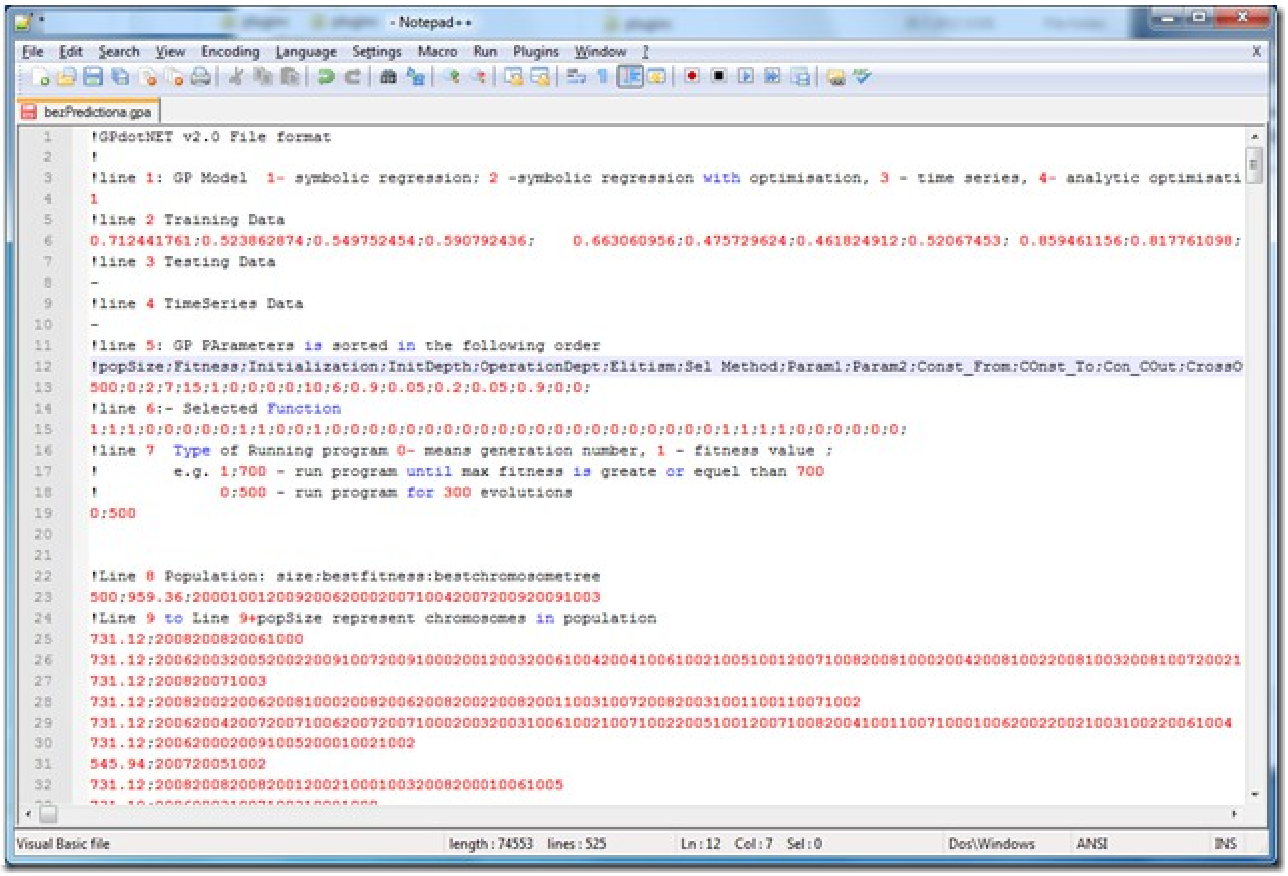
Deleting the nodes in function

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.

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.


- Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
bezPredictiona.gpa
Visual Basic file
length : 74553 lines : 525
Ln : 12 Col : 7 Sel : 0
Dos\Windows
ANSI
INS

```
!GPdotNET v2.0 File format
!
!line 1: GP Model  1- symbolic regression; 2 -symbolic regression with optimisation, 3 - time series, 4- analytic optimisati
1
!line 2 Training Data
0.712441761;0.523862874;0.549752454;0.590792436;    0.663060956;0.475729624;0.461824912;0.52067453; 0.859461156;0.817761098;
!line 3 Testing Data
-
!line 4 TimeSeries Data
-
!line 5: GP PArameters is sorted in the following order
!popSize;Fitness;Initialization;InitDepth;OperationDept;Elitism;Sel Method;Param1;Param2;Const_From;COnst_To;Con_COut;CrossO
500;0;2;7;15;1;0;0;0;0;10;6;0.9;0.05;0.2;0.05;0.9;0;0;
!line 6:- Selected Function
1;1;1;0;0;0;0;0;1;1;0;0;1;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;1;1;1;1;0;0;0;0;0;0;
!line 7  Type of Running program 0- means generation number, 1 - fitness value ;
!         e.g. 1;700 - run program until max fitness is greate or equel than 700
!               0;500 - run program for 300 evolutions
0;500


!Line 8 Population: size;bestfitness;bestchromosometree
500;959.36;20001001200920062000200710042007200920091003
!Line 9 to Line 9+popSize represent chromosomes in population
731.12;2008200820061000
731.12;200620032005200220091007200910002001200320061004200410061002100510012007100820081000200420081002200810032008100720021
731.12;200820071003
731.12;2008200220062008100020082006200820022008200110031007200820031001100110071002
731.12;20062004200720071006200720071000200320031006100210071002200510012007100820041001100710001006200220021003100220061004
731.12;2006200020091005200010021002
545.94;200720051002
731.12;20082008200820012002100010032008200010061005
```

Figure 13: GPdotNET file format highlighted in Notepad ++