

```
>>gp=runGP(@Y11_config);  
Press a key to continue
```

GPTIPS 2

Symbolic data mining platform for MATLAB
Copyright (C) Dominic Searson 2009-2015

Contact: searson@gmail.com

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 3 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: <http://www.gnu.org/licenses>

Run parameters

Population size: 100
Number of generations: 100
Number of runs: 1
Parallel mode : off
Tournament type: regular
Tournament size: 15
Elite fraction: 0.3
Fitness cache: enabled
Lexicographic selection: True
Max tree depth: 4
Max nodes per tree: Inf
Using function set: TIMES MINUS PLUS
Number of inputs: 5
Max genes: 12
Constants range: [-10 10]
Complexity measure: expressional
Fitness function: regressmulti_fitfun.m

Generation 0

Best fitness: 0.10856
Mean fitness: 0.22421
Best complexity: 204
Inputs in best individual: x1 x2 x3 x4 x5

Generation 20

Best fitness: 0.097293

Mean fitness: 0.10964
Best complexity: 226
Inputs in best individual: x1 x2 x3 x4 x5

Generation 40
Best fitness: 0.097049
Mean fitness: 0.10657
Best complexity: 220
Inputs in best individual: x1 x2 x3 x4 x5

Generation 60
Best fitness: 0.096376
Mean fitness: 0.1031
Best complexity: 198
Inputs in best individual: x1 x2 x3 x4 x5

Generation 80
Best fitness: 0.096335
Mean fitness: 0.10305
Best complexity: 214
Inputs in best individual: x1 x2 x3 x4 x5

Finalising run.
GPTIPS run complete in 0.56 min.
Best fitness acheived: 0.096159

Evaluate the best individual of
the runs on the fitness function using:
>>runtree(gp,'best');
Press a key to continue

Next, use the the GPPRETTY command on the best individual:
>>gppretty(gp,'best')
Press a key to continue

Simplified genes

Gene 1 and bias term

0.03322 x1 + 0.03322 x2 + 2.578

Gene 2

-0.00002977 x4 (x2 + x3) (2.0 x1 + x2)

Gene 3

$$- 0.00001482 x_1 (x_2 x_4^2 + x_2 + x_3)$$

Gene 4

$$0.2836 x_1$$

Gene 5

$$0.03605 x_5 (x_3 + 1.0)$$

Gene 6

$$-1.309 x_3$$

Gene 7

$$0.03528 x_1 + 0.01764 x_2 - 0.01764 (x_1 - 1.0 x_3) (x_3 - 1.0 x_4)$$

Gene 8

$$0.4097 x_3 - 0.4097 x_4$$

Gene 9

$$- 0.08421 x_2 - 0.08421 x_3 x_5$$

Gene 10

$$0.03516 x_4 (x_2 + x_3) - 0.07032 x_3 (x_1 - 1.0 x_5)$$

Gene 11

$$-0.02633 x_2 x_4$$

Gene 12

$$-0.00003858 x_4 (x_2 + x_3) (x_3 + x_5)$$

Simplified overall GP expression

$$\begin{aligned} & 0.3521 x_1 - 0.03335 x_2 - 0.8992 x_3 - 0.4097 x_4 + 0.03605 x_5 - 0.00001482 x_1 (x_2 + x_3) + \\ & 0.03516 x_4 (x_2 + x_3) - 0.01764 (x_1 - 1.0 x_3) (x_3 - 1.0 x_4) - 0.07032 x_3 (x_1 - 1.0 x_5) - \\ & 0.02633 x_2 x_4 - 0.04817 x_3 x_5 - 0.00002977 x_4 (x_2 + x_3) (2.0 x_1 + x_2) - \\ & 0.00001482 x_1 x_2 x_4^2 - 0.00003858 x_4 (x_2 + x_3) (x_3 + x_5) + 2.578 \end{aligned}$$

Next, use the the DRAWTREE command:

```
>>drawtrees(gp,'best')
```

Press a key to continue

Trees drawn to trees.htm

Opening in system browser.

Finally, an HTML report listing the models on the Pareto optimal front of model expressional complexity and performance can be generated using the PARETOREPORT function.

```
>>paretoreport(gp)
```

Press a key to continue

100 models passed R^2 training (≥ 0) and expressional complexity ($\leq \text{Inf}$) filter ...

Computing pareto front on training data...

Removing genotype duplicates from 19 remaining models ...

9 models passed the filtering process.

Model report created in pareto.htm

Opening report in system browser.

end

```
>>
```