Repeated Sequences in Genetic Programming

W. B. Langdon

Computer Science

University of Essex

29.6.2012

Introduction

- Langdon + Banzhaf in Memorial University, Canada
- Emergence: Repeated Sequences
- Repeated Sequences in Biology
- Linear and Tree Genetic Programming
- Test problems
- Repeated sequences, fragments and subtrees
- Movies
- So what?
  - Where does this lead next?
  - Other emergent phenomena?
- Conclusions

Emergence

- Emergence of effects that have not been explicitly programmed into the system.
- Simple rules lead to complex behaviour. Intelligence emerging from many trivial interactions.
- Particle Swarm Optimisation (PSO)
  - Flocking
  - Boids
  - Swarm intelligence

- Genetic Programming
  - Bloat
  - Repeated Sequences

Repeats in DNA

- Many different types of repeated DNA sequence. Classified by repeat sequence length, number of repeats, location in DNA molecule etc. etc.
  - Some may have biological meaning, e.g. as a clock counting cell divisions and enforcing limit, cell life limited, so cancer prevented.
  - Repeated sequences in both expressed (protein coding) and non-expressed DNA.
- DNA whose sequence is not maintained by selection will develop periodicities as a result of random crossover [G.P. Smith, 1976].

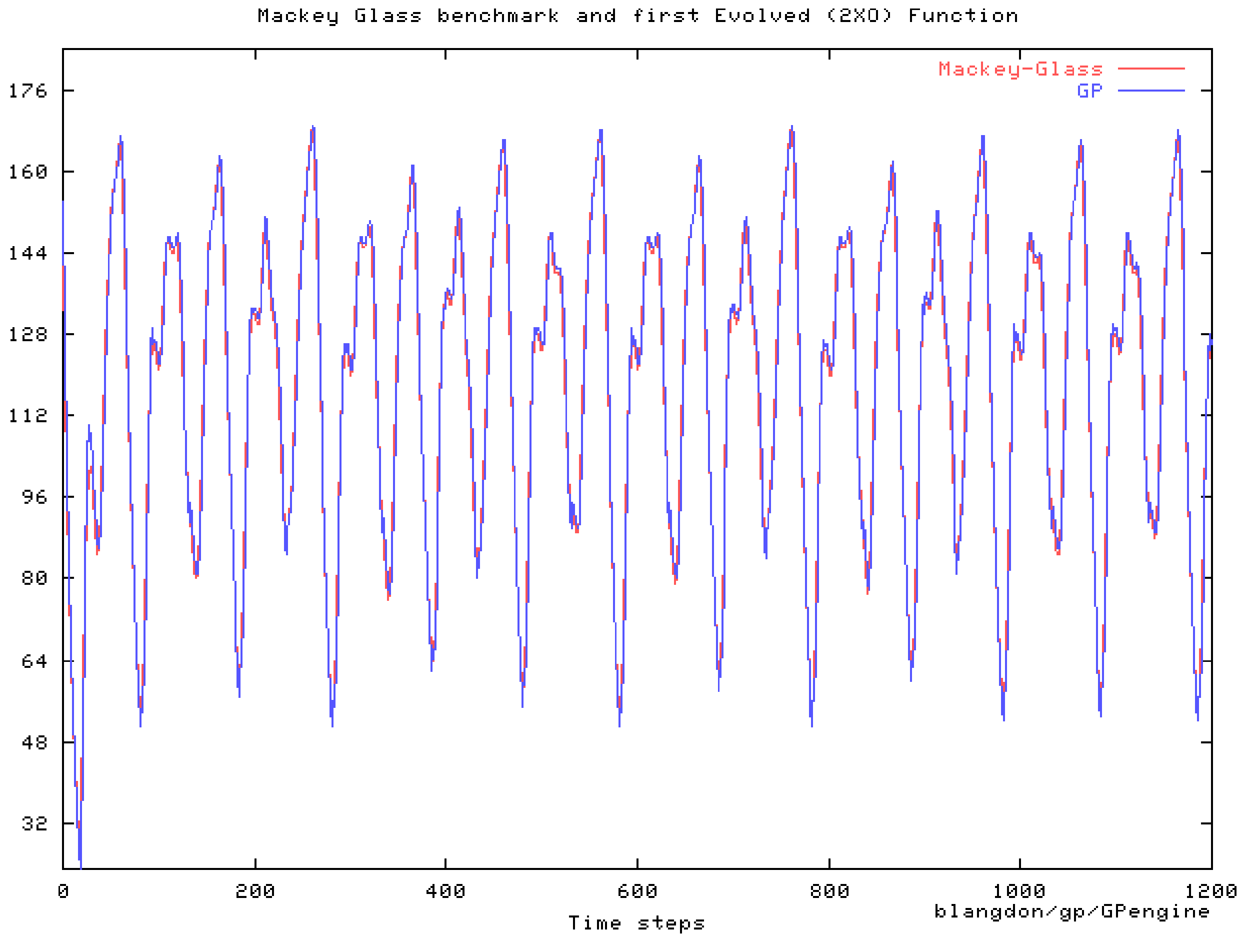
Demonstration problems

- Want to run GP for many generations. Hard problems, not immediately solved.
- Want range of different problems
  - Time series modeling. One variable, short integers (byte) arithmetic
  - Bioinformatics. Binary classification, floating point, 20 inputs.

Mackey-Glass Chaotic Time Series

- Hard (impossible) since chaotic time series.
- IEEE benchmark, 1201 data points.
- Fast signal processing (integer arithmetic)
- 7 time lags: 1, 2, 4, …, 128 steps ago.

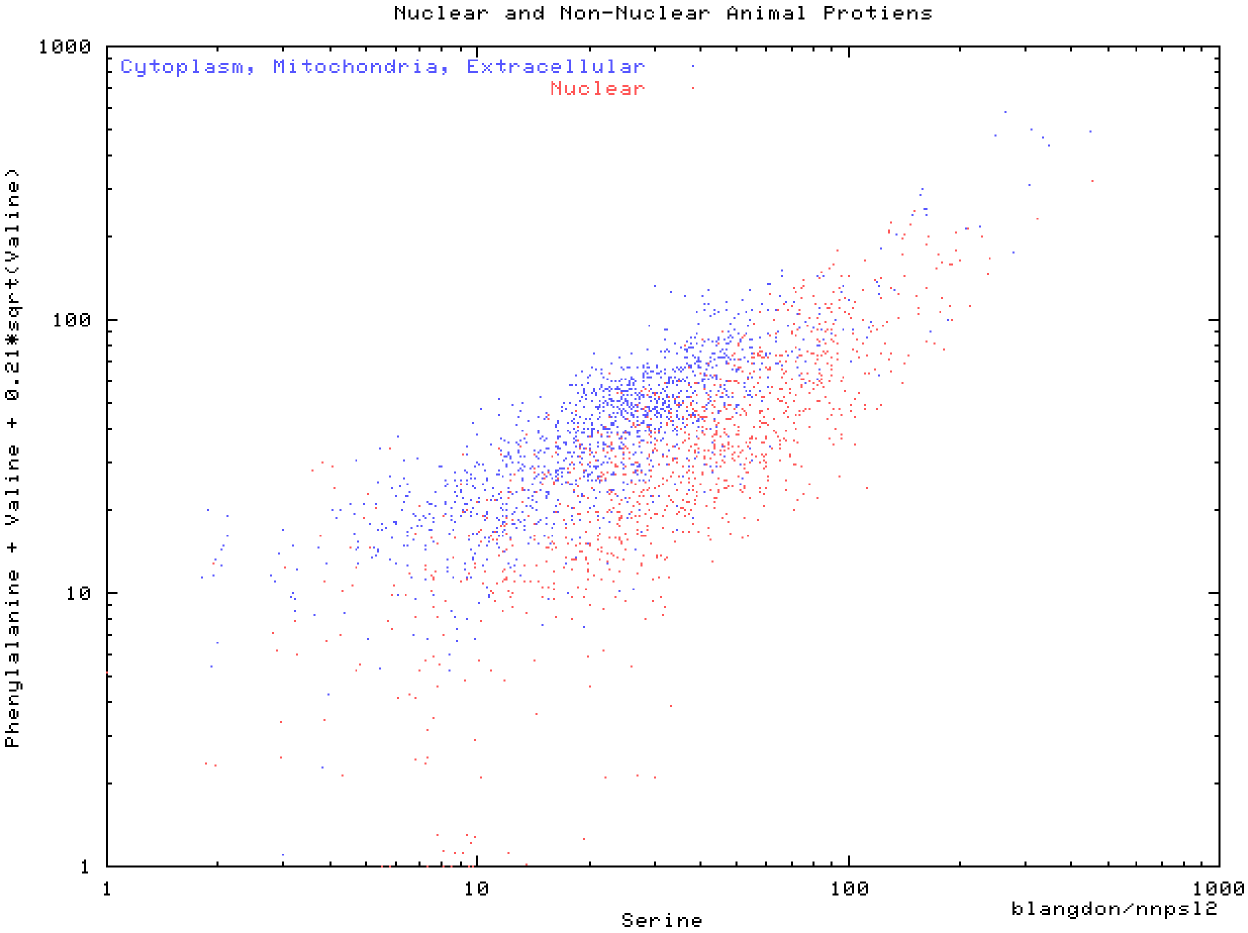
Mackey-Glass



Predicting Protein Location

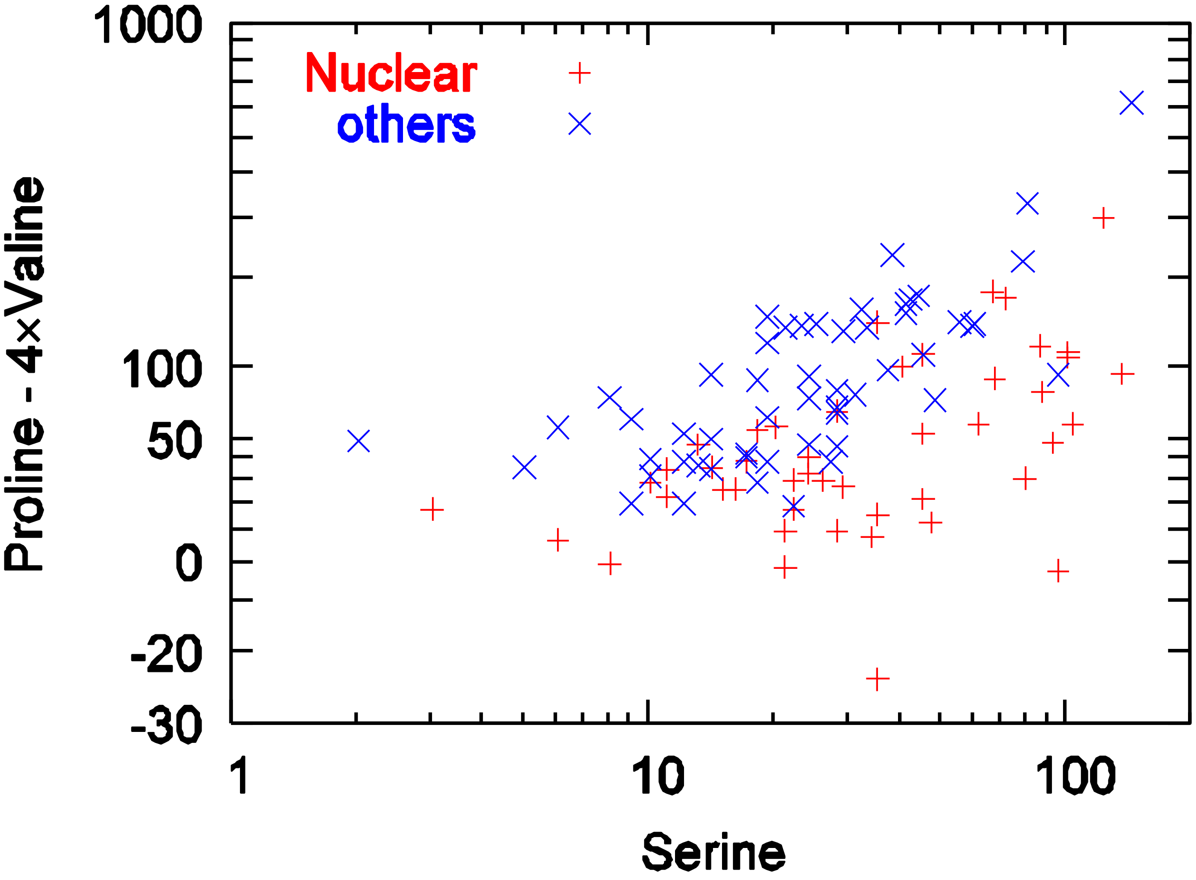
- Given only number of each amino acid (i.e. cheap info, Swissprot) in a protein, predict what it is. Very hard.
- Easier: predict where the protein will be found
  - Simplified (A. Reinhardt and T. Hubbard, 1998) which covers animals and microbes, to just animals and two classes: In the cell nucleus or not.

Animal Nuclear Proteins



Non-linear 2D projection from 20 Dimensional Space

Animal Nuclear Proteins



Non-linear 2D projection from 20 Dimensional Space

Genetic Programming Approaches

- Linear GPengine (Nordin)
  - crossover with mutation
  - Headless chicken mutation (HCX) only
- Linear Machine Code Discipulus
- Tree GP

Linear Genetic Programming

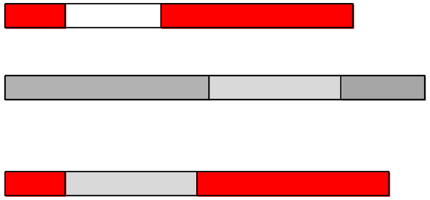
- Chromosome is program.
  - A linear sequence instructions
  - Executed from start to end (no loops)
- GPengine - interpreted.
- Discipulus Intel 486 instructions

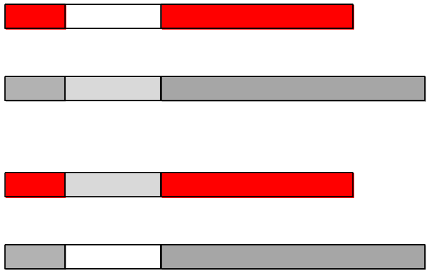
Linear GP Chromosome

- GPengine instruction format

Output	Arg 1	Opcode	Arg 2
R0..R7	R0..R7	+ - * /	0...127 or R0..R7

- 90% Crossover
- 40% Mutation. Pop 500.
- Two point (4 crossover chosen independently)
- Homologous (parent crossover points aligned)





Performance (all approaches solve problems)

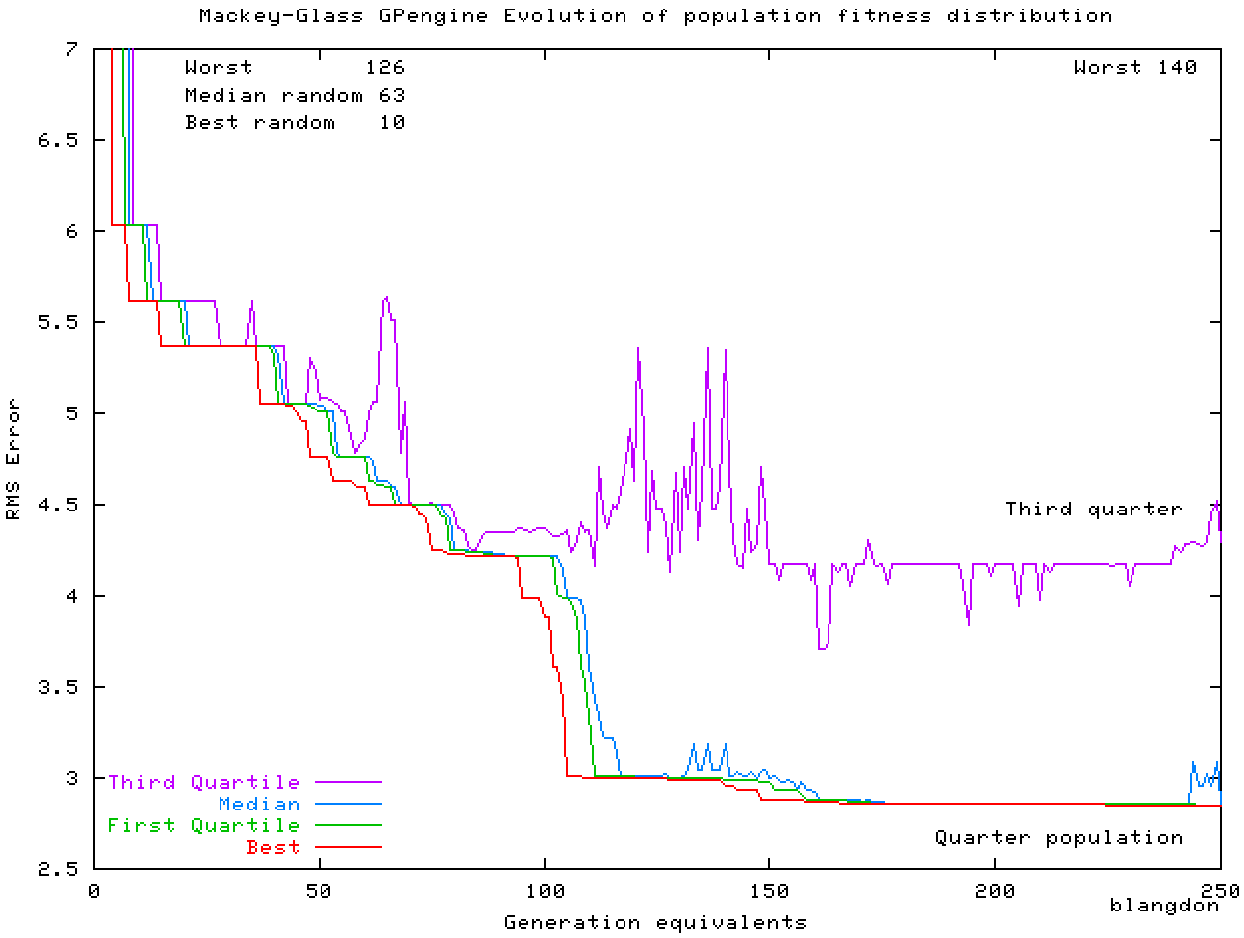
Predicting M-G chaotic Time Series

	RMS error	Mean
Linear GP	1.6-5.4	3.8
Tree GP	1.1-4.9	3.5

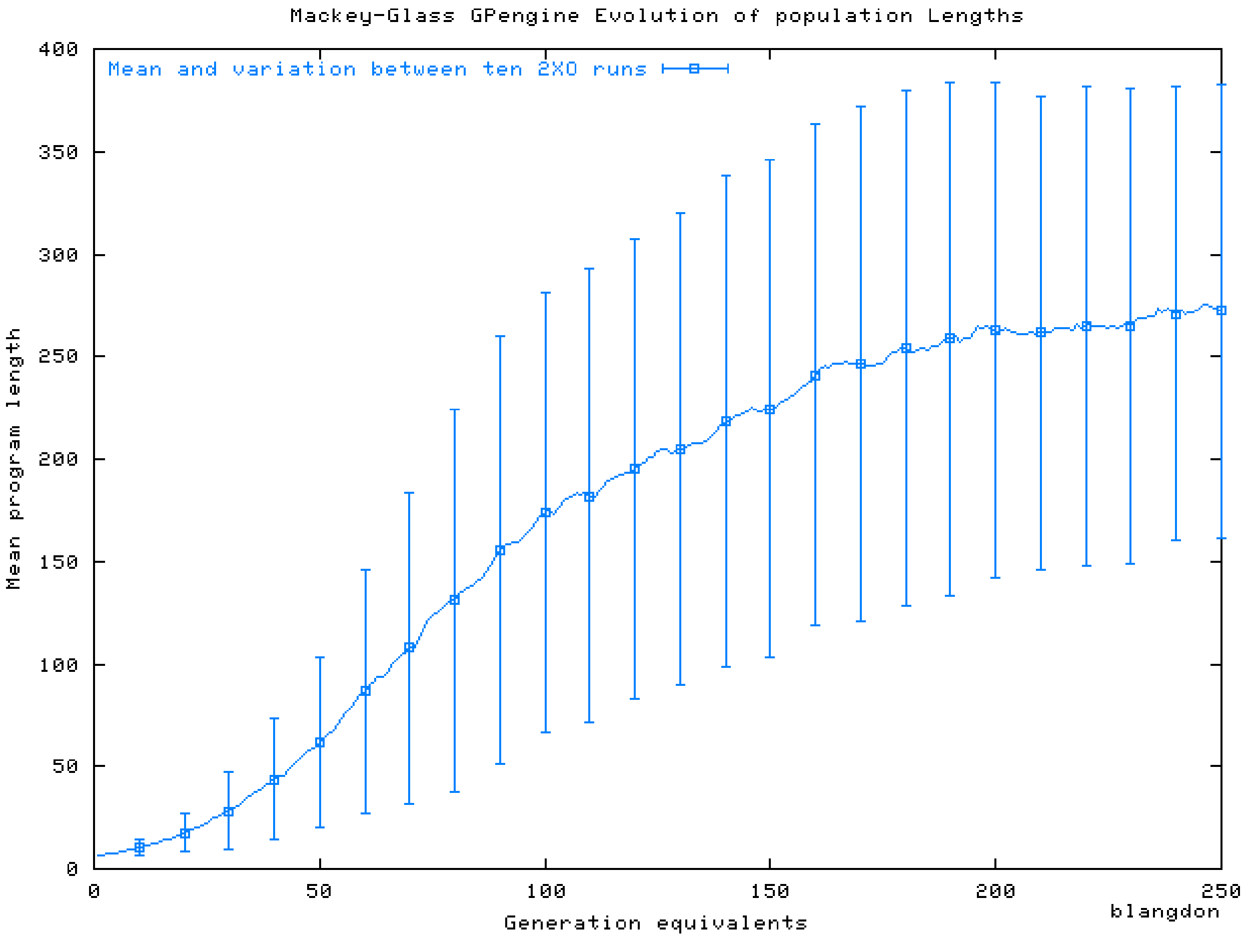
Nuclear Protein prediction (holdout set)

Discipulus	78-82%	80%
Tree GP	78-83%	81%

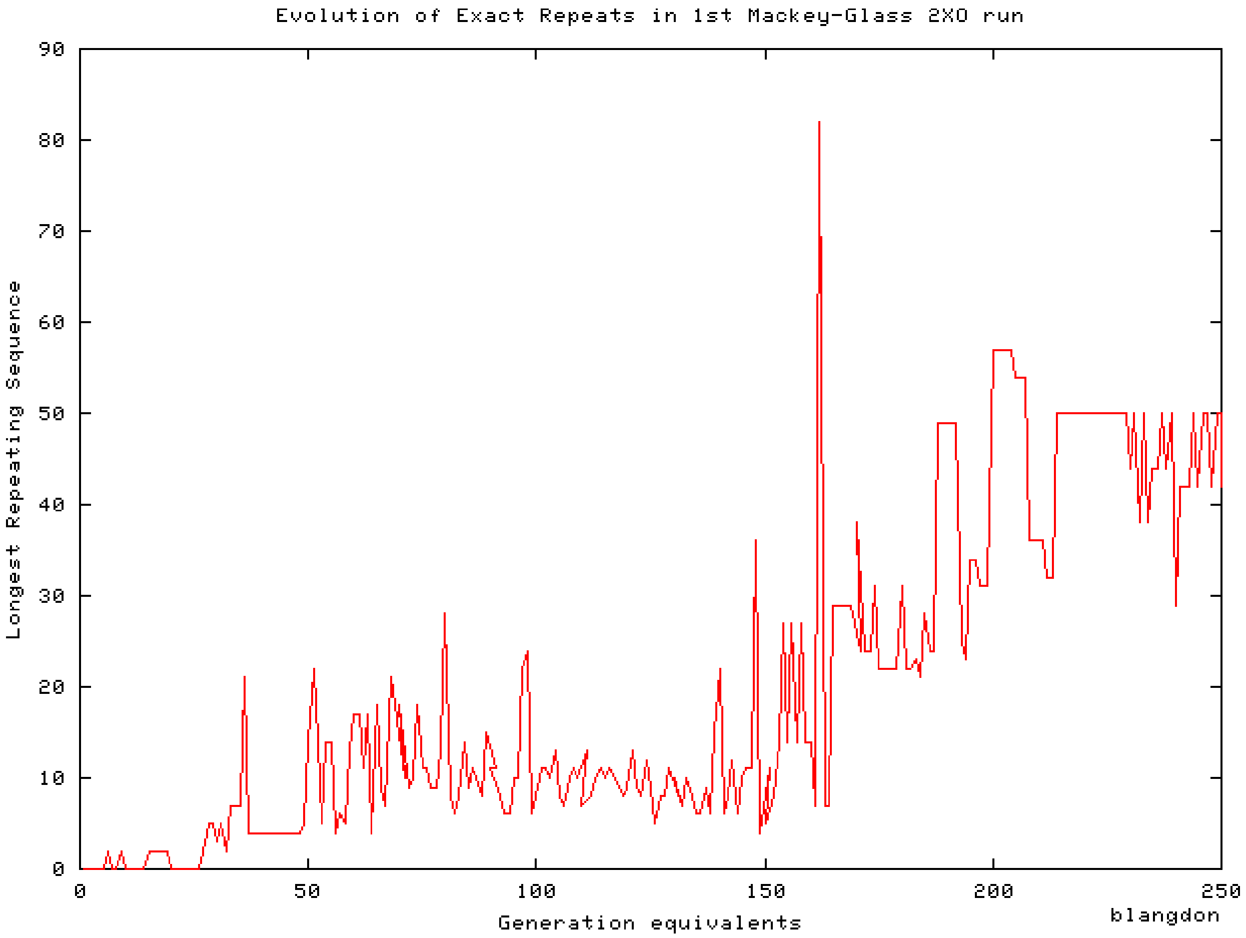
Evolution of Mackey-Glass error



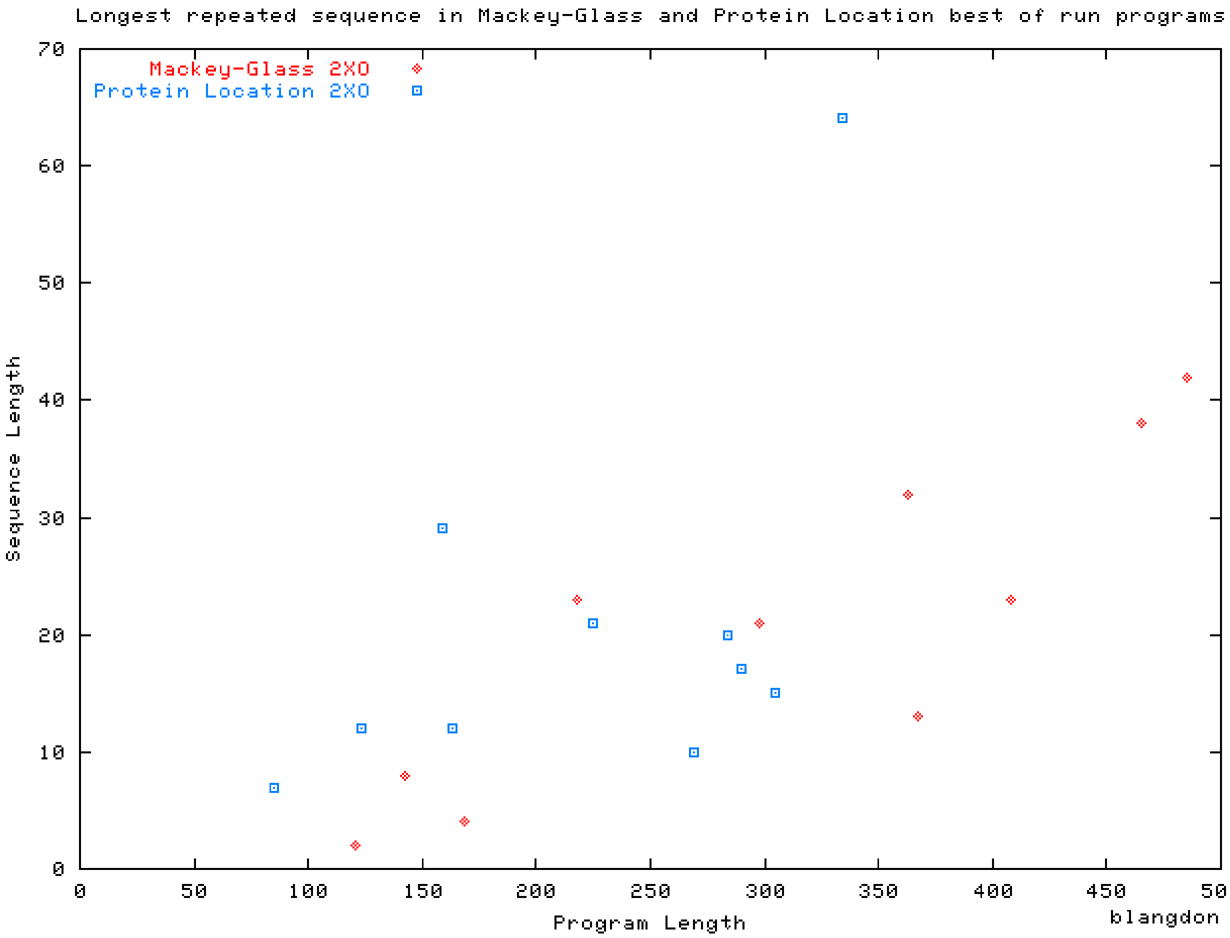
Evolution of M-G program length

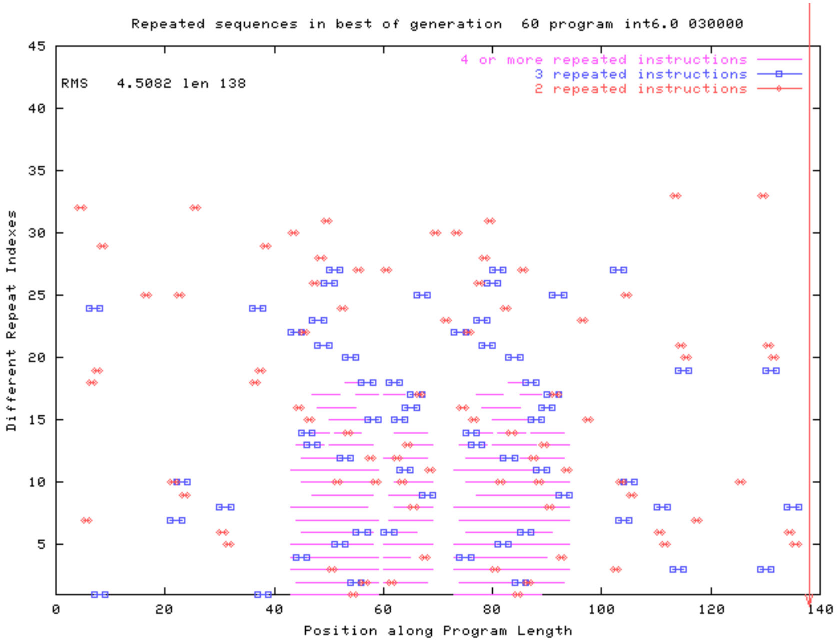


Length of Repeated Sequences



Longest Repeats M-G and Protein

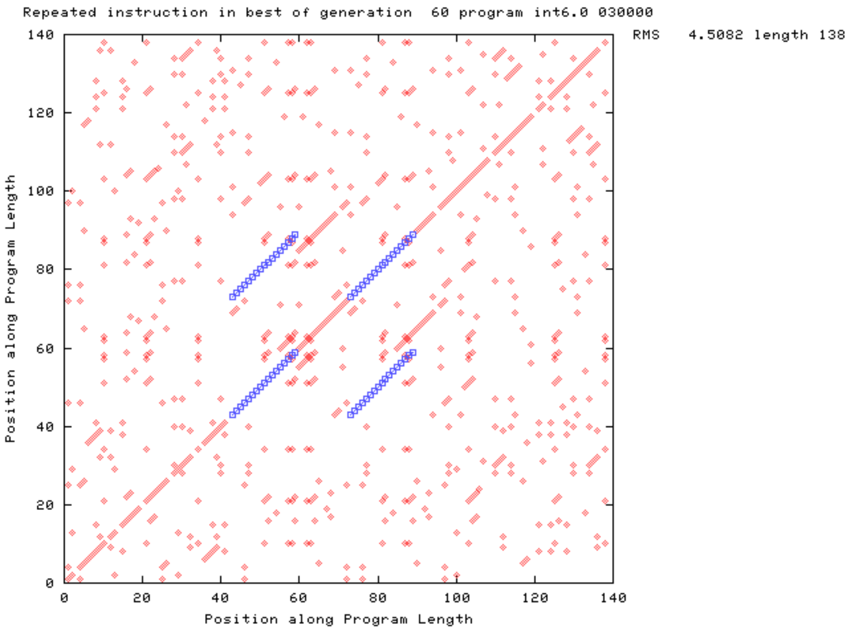




- Red arrow indicates length of program.
- Single repeated instructions are not shown.
- Repeated pairs of instructions are shown in red.
- Repeated sequence of 3 instructions in blue.
- Four or more are plotted with purple lines.
- Length and Fitness, RMS error, as numbers.

Evolution of Location of Repeated Instructions

- First two point crossover Mackey-Glass GPengine run
  - http://www.cs.ucl.ac.uk/staff/W.Langdon/gecco2004lb/int6.0.all.rep2_movie.gif



- Dot at i,j means instruction at location i is identical to that at location j.
- 1-10 repeated instructions are shown with red.
- 11 or more repeated sequence shown in blue.
- Length and Fitness, RMS error, given numerically.
- Same Mackey-Glass 2point crossover run

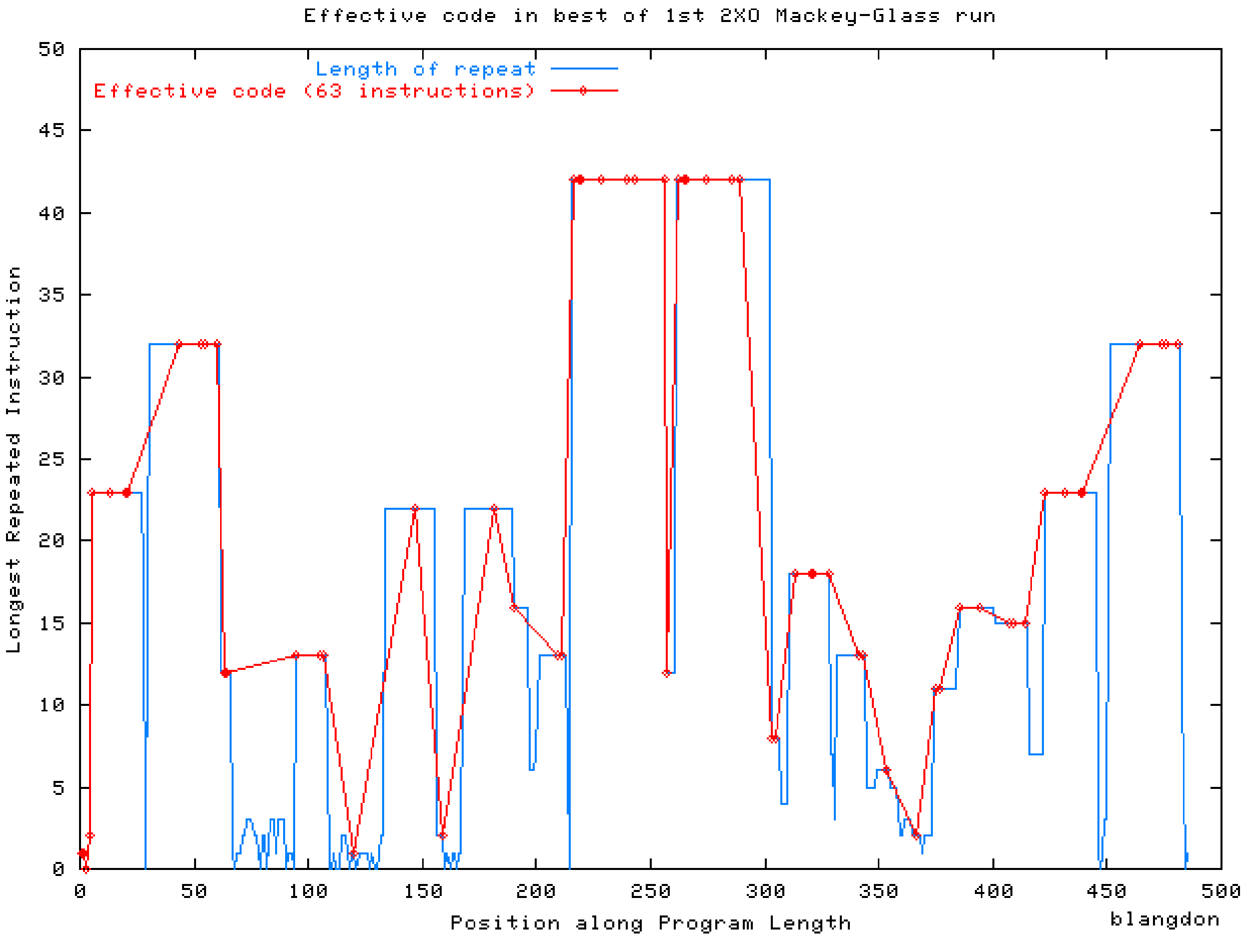
Animation

- 250 generations Mackey-Glass GPengine
  - http://www.cs.ucl.ac.uk/staff/W.Langdon/gecco2004lb/int6.0.250.movie.gif

Effective Code

- Majority of instructions have no effect on the output of the programs.
- No obvious link between repeat and effectiveness

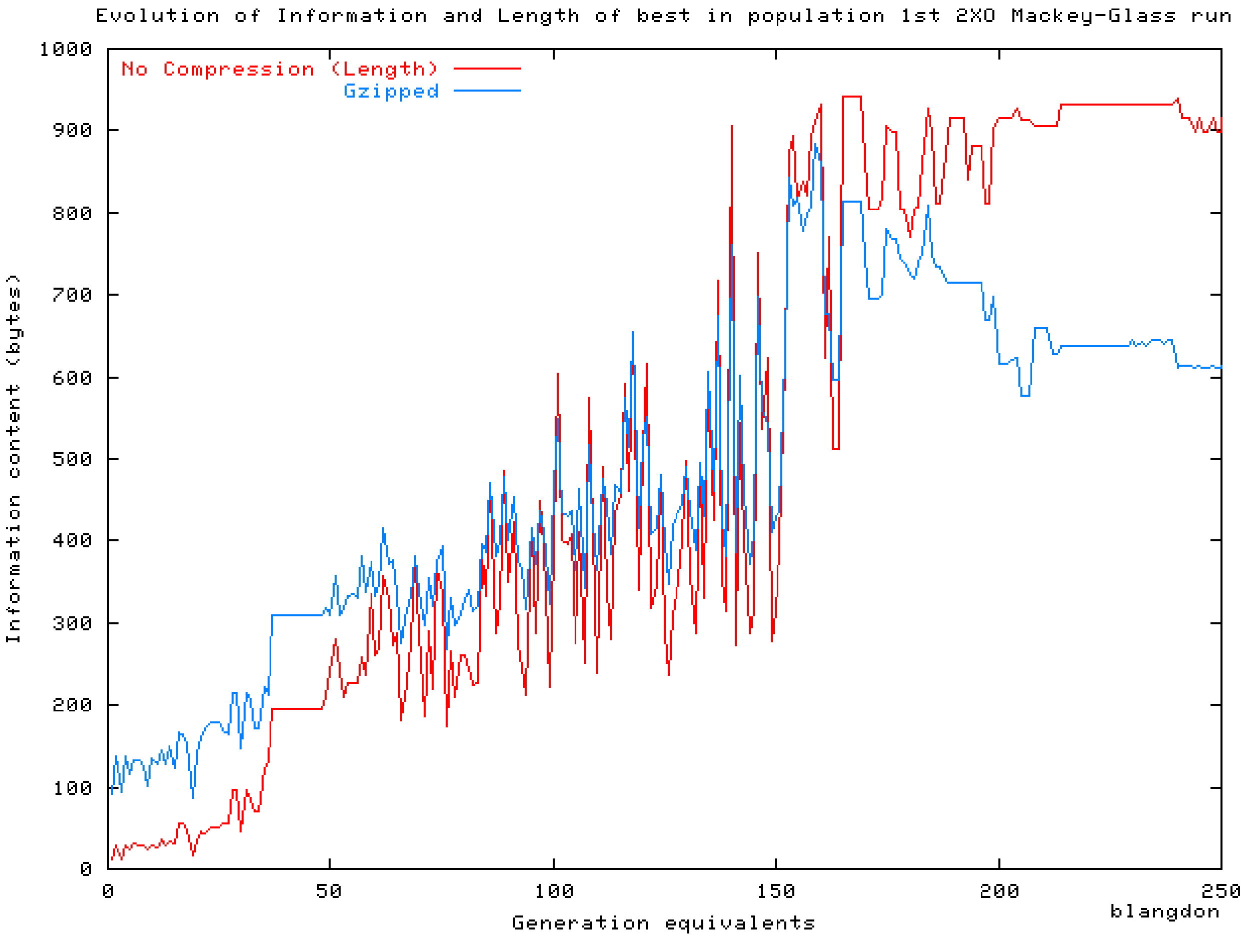
Introns and Repeats evolved in one Mackey-Glass program



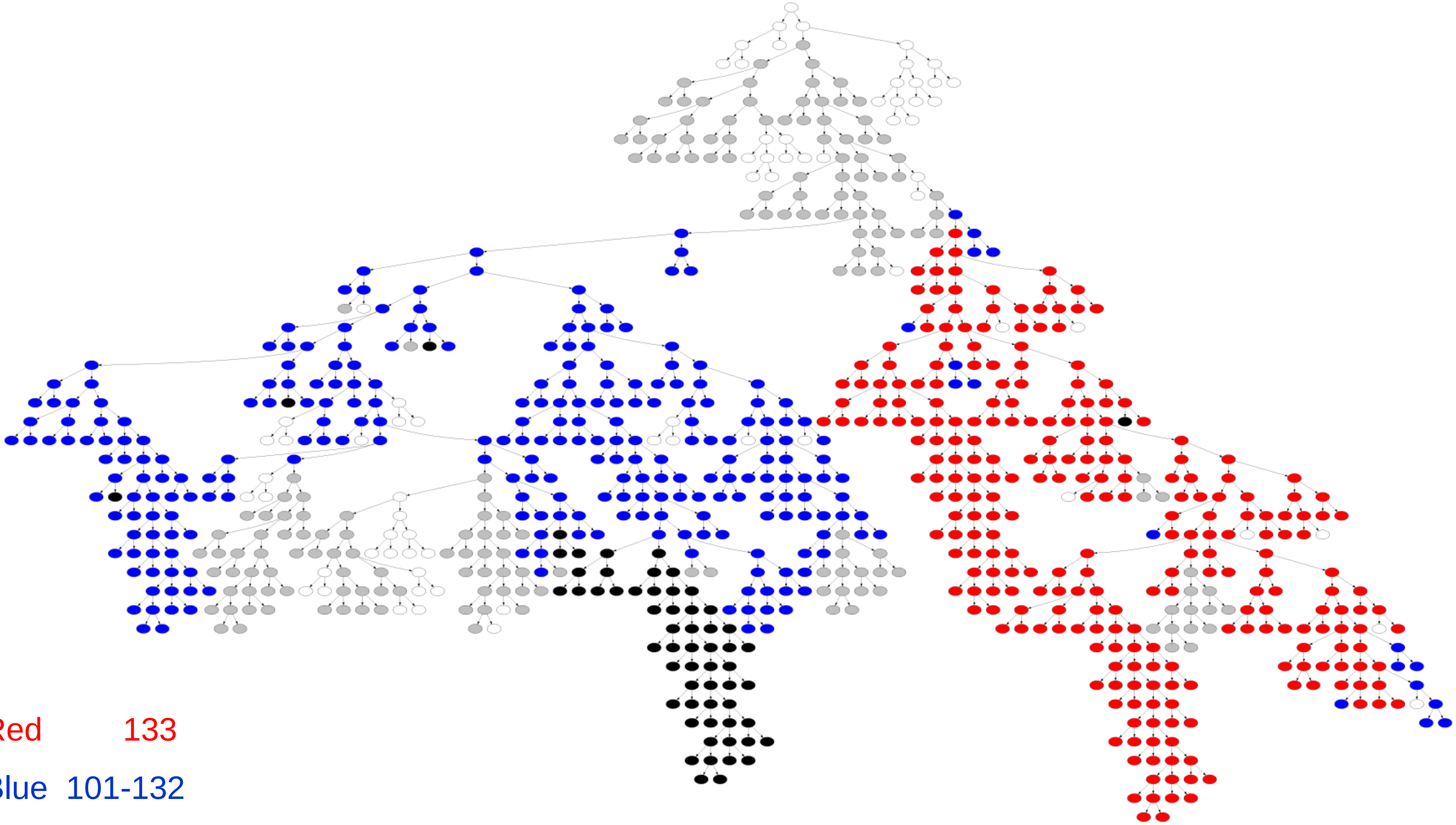
Information Content

- Lempel-Ziv compression shows bloated programs' contain less information than random program of same length.

Evolution of Information Content



Repeats in largest Protein Prediction program



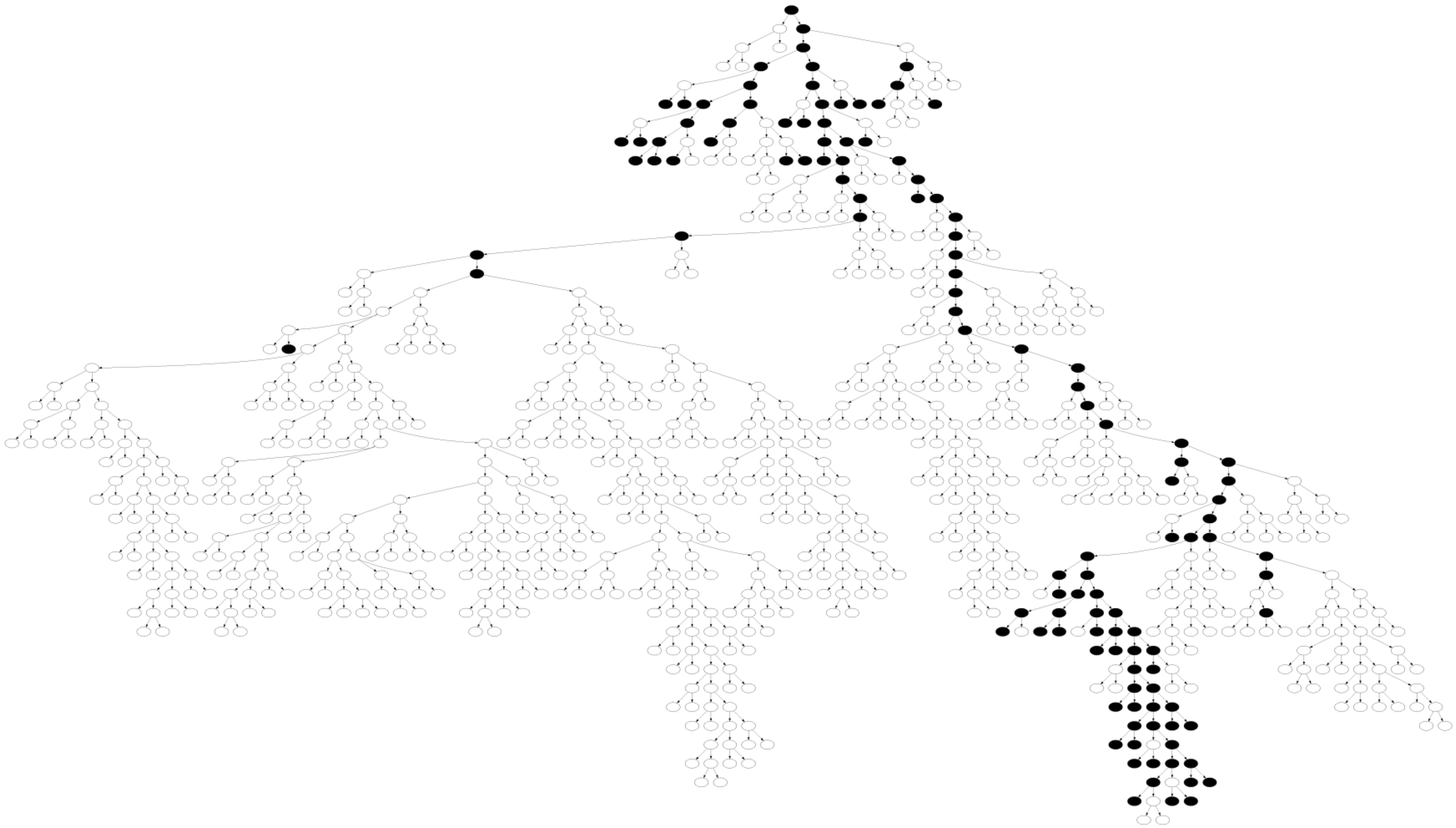
Red 133

Blue 101-132

Black 33-100

Grey 11-32

Important Nodes



Black changes >10 training cases

Discussion

- In trees, can get *diffuse introns* whereby whole program depends only on fraction of tree. Not classic introns, since most functions do depend on both arguments.
- Crossover evolves trees similar fractal shape properties as random trees BUT
- Repeats not random.
- Many subtrees have high fitness and pass information towards root, BUT
- Much of program can be discarded with little impact on fitness
- Genetic programming on simple problems assembles complete solutions by gradually, randomly, reusing existing partial solutions to get small improvements, rendering existing parts less important.

Conclusions

- On different problems and different GPs (2 linear and tree) where length is not constrained, repeated sequences/subtrees/fragments emerge from crossover
- Repeats cover large fraction of fit programs.
- This is an example of *emergence*.
  - Are there examples in your EA of effects (which were not pre-programmed) which *spontaneously evolved?*

More information

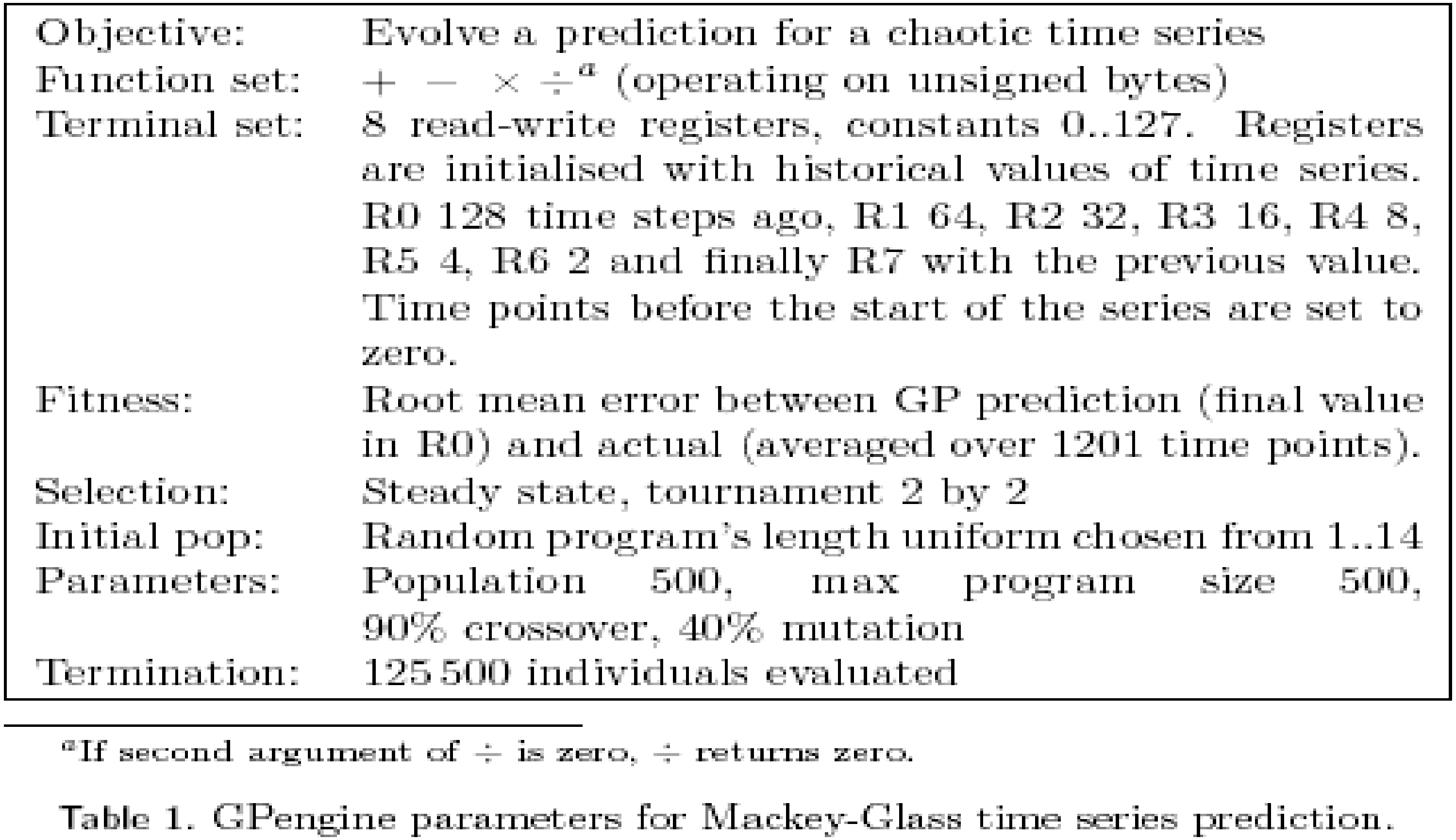
References:
Repeated Sequences in Linear GP Genomes, W.B. Langdon and W. Banzhaf, (GECCO'2004 late breaking paper PDF gzipped postscript). Movie. Poster

Smith, G.P. (1976) "Evolution of Repeated DNA Sequences by Unequal Crossover." *Science*, 191(4227), 528-535. [PDF]).

More information on GP

- http://www.cs.ucl.ac.uk/staff/W.Langdon/
  - *Foundations of GP*, Springer, 2002
  - *GP and Data Structures*, Kluwer, 1998
- http://liinwww.ira.uka.de/**bibliography**/Ai/genetic.programming.html
- http://www.cs.ucl.ac.uk/staff/W.Langdon/**lisp2dot**.html

GPengine Mackey-Glass

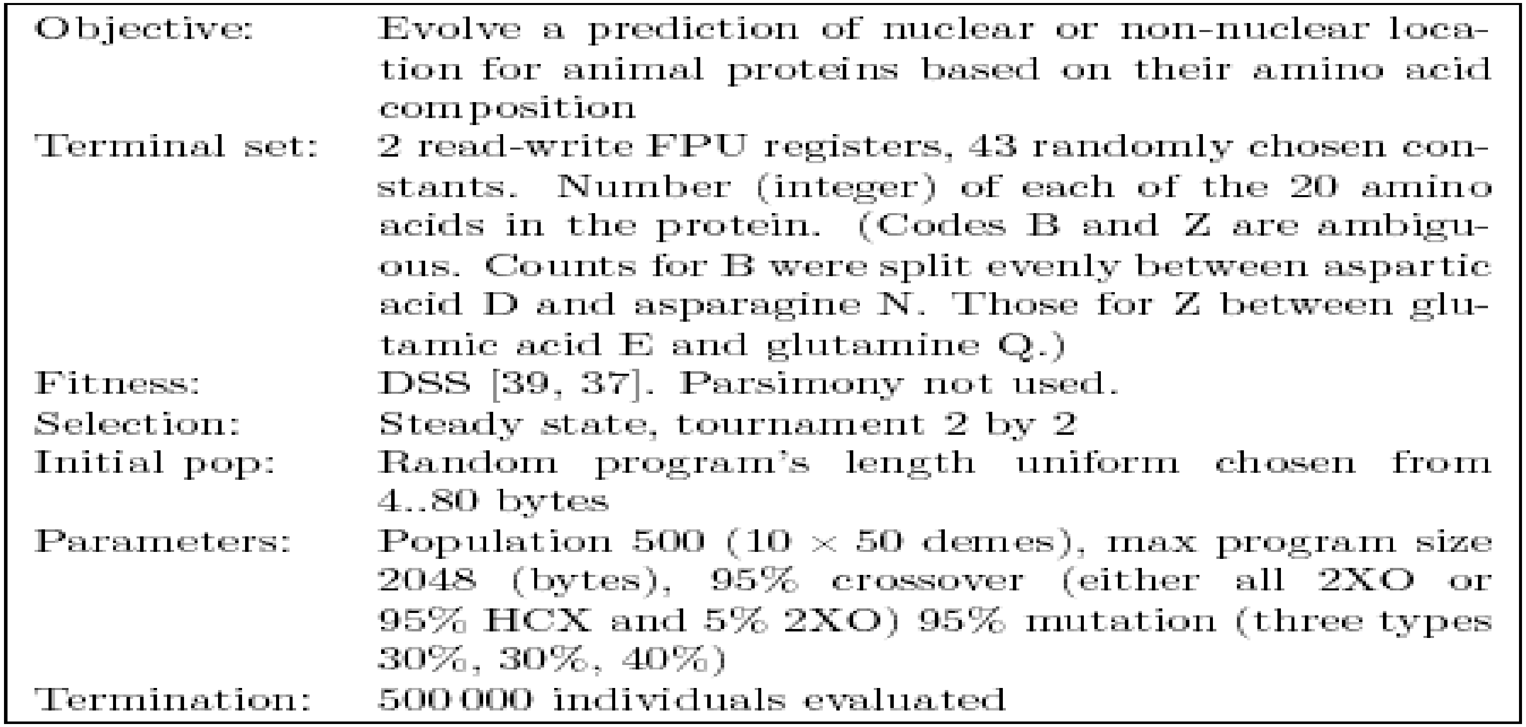


Objective:	Evolve a prediction for a chaotic time series
Function set:	$+ \ - \ \times \ \div$[a] (operating on unsigned bytes)
Terminal set:	8 read-write registers, constants 0..127. Registers are initialised with historical values of time series. R0 128 time steps ago, R1 64, R2 32, R3 16, R4 8, R5 4, R6 2 and finally R7 with the previous value. Time points before the start of the series are set to zero.
Fitness:	Root mean error between GP prediction (final value in R0) and actual (averaged over 1201 time points).
Selection:	Steady state, tournament 2 by 2
Initial pop:	Random program's length uniform chosen from 1..14
Parameters:	Population 500, max program size 500, 90% crossover, 40% mutation
Termination:	125 500 individuals evaluated

[a] If second argument of ÷ is zero, ÷ returns zero.

Table 1. GPengine parameters for Mackey-Glass time series prediction.

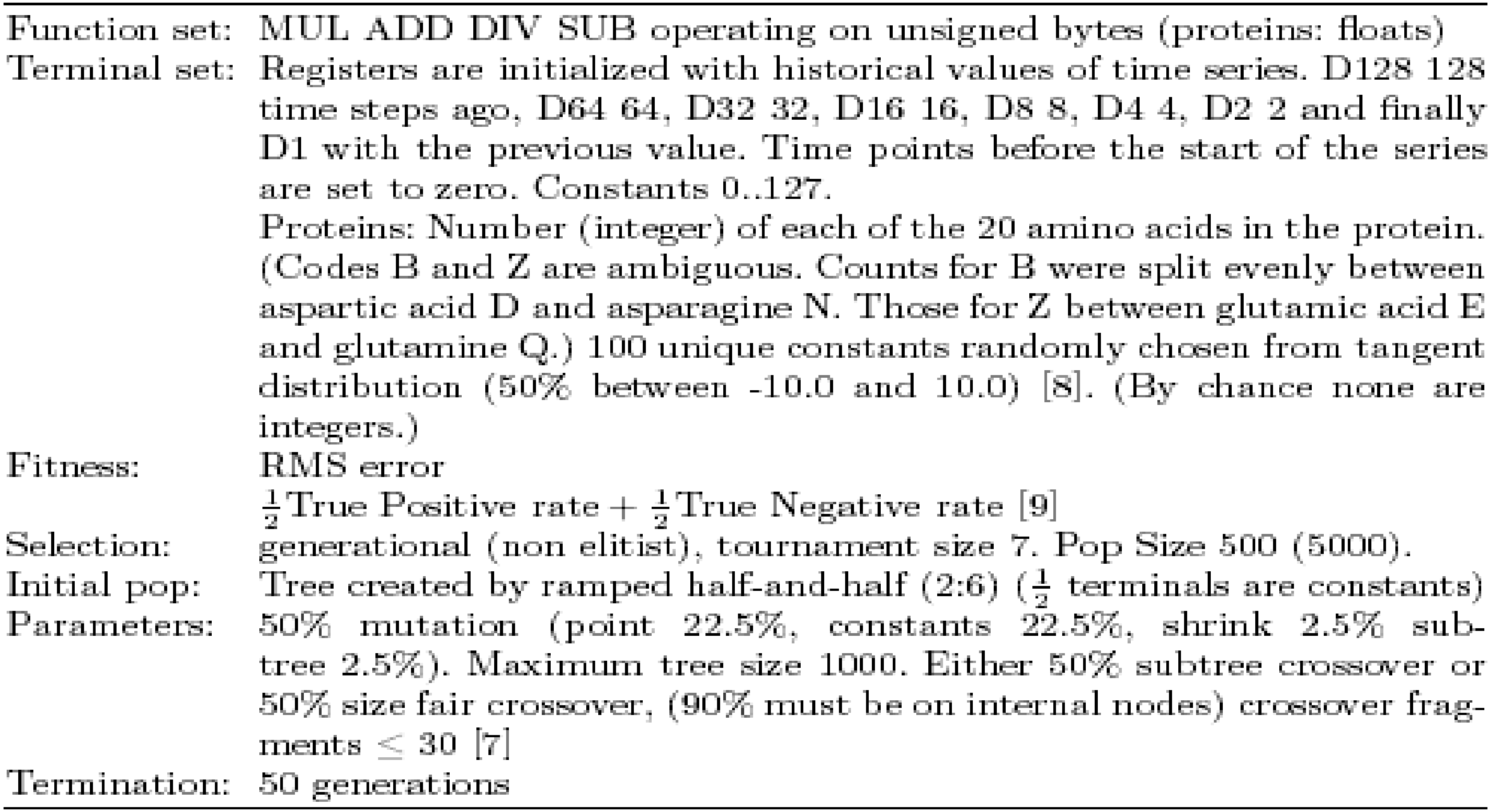
Discipulus Protein Prediction



Objective:	Evolve a prediction of nuclear or non-nuclear location for animal proteins based on their amino acid composition
Terminal set:	2 read-write FPU registers, 43 randomly chosen constants. Number (integer) of each of the 20 amino acids in the protein. (Codes B and Z are ambiguous. Counts for B were split evenly between aspartic acid D and asparagine N. Those for Z between glutamic acid E and glutamine Q.)
Fitness:	DSS [39, 37]. Parsimony not used.
Selection:	Steady state, tournament 2 by 2
Initial pop:	Random program's length uniform chosen from 4..80 bytes
Parameters:	Population 500 (10 × 50 demes), max program size 2048 (bytes), 95% crossover (either all 2XO or 95% HCX and 5% 2XO) 95% mutation (three types 30%, 30%, 40%)
Termination:	500 000 individuals evaluated

Table 3. Discipulus parameters used in animal protein location prediction experiments. Only the maxiumnum program size and HCX were changed from factory defaults.

Tree Mackey-Glass (Protein Localisation)



Function set:	MUL ADD DIV SUB operating on unsigned bytes (proteins: floats)
Terminal set:	Registers are initialized with historical values of time series. D128 128 time steps ago, D64 64, D32 32, D16 16, D8 8, D4 4, D2 2 and finally D1 with the previous value. Time points before the start of the series are set to zero. Constants 0..127. Proteins: Number (integer) of each of the 20 amino acids in the protein. (Codes B and Z are ambiguous. Counts for B were split evenly between aspartic acid D and asparagine N. Those for Z between glutamic acid E and glutamine Q.) 100 unique constants randomly chosen from tangent distribution (50% between -10.0 and 10.0) [8]. (By chance none are integers.)
Fitness:	RMS error $\frac{1}{2}$True Positive rate + $\frac{1}{2}$True Negative rate [9]
Selection:	generational (non elitist), tournament size 7. Pop Size 500 (5000).
Initial pop:	Tree created by ramped half-and-half (2:6) ($\frac{1}{2}$ terminals are constants)
Parameters:	50% mutation (point 22.5%, constants 22.5%, shrink 2.5% subtree 2.5%). Maximum tree size 1000. Either 50% subtree crossover or 50% size fair crossover, (90% must be on internal nodes) crossover fragments $\leq$ 30 [7]
Termination:	50 generations