GECCO 2014 "Hot of the Press" track, Christian Igel and Dirk Arnold *Eds.*, Vancouver, 12-16 July.

Improving Source Code with Genetic Programming

William B. Langdon and Mark Harman
Dept. of Computer Science, University College London Gower Street, WC1E 6BT, UK
Code etc: http://www.cs.ucl.ac.uk/staff/W.Langdon/ftp/gp-code/bowtie2gp/

ABSTRACT

"Optimising Existing Software with Genetic Programming" (to appear in IEEE Transactions on Evolutionary Computation doi:10.1109/TEVC.2013.2281544) describes a recent experiment in which a state-of-the-art Bioinformatics program was manipulated by GP to give a variant automatically tuned to a particular task. The program consists of 50 000 lines of C++. Evolution was able to find a change which speeds it up on the chosen task on average by a factor of 70, yet still give good answers, indeed the results are slightly better.

GISMOE uses a BNF grammar specific to Bowtie2 when mutating genetically improved programs (GIP). Patches delete, move or insert existing lines of code. No new code is created. Mutants' fitness is measured by running them using DNA from The Thousand Genomes Project.

Categories and Subject Descriptors I.2.8 [search]: heuristic I.2.2[Artificial Intelligence]: Automatic Programming D.2.8 [Software Engineering]: Metrics *performance measures*

Keywords

automatic software re-engineering, Bowtie2^{GP}, multiple objective exploration, search based software engineering (SBSE).

Summary

Bowtie2 was written by Ben Langmead. It is the state-of-the-art open source tool for looking up short nextGen DNA sequences in the human genome. GP (Figure 1) uses variable length linear chromosomes, each gene of which specifies one mutation to one line of code. During evolution approximately 2/3 of mutants compile and run successfully.

Each generation five DNA sequences, covering a range of difficulty, are chosen for fitness testing. Each Bowtie2 mutant is run (subject to a CPU limit) on all five and its answers compared with those given by the original code. Only mutants which compile and which are better may have children. The top half of the population are selected to have two children each. If less than half the population (10) are better than Bowtie 2, any missing children are created at random. Quality is given by average Smith-Waterman score.

Crossover creates a child by appending two patch lists. Mutation appends another one line change. Mutations are random but are heavily weighted towards lines of code which are either used many times or which scale badly.

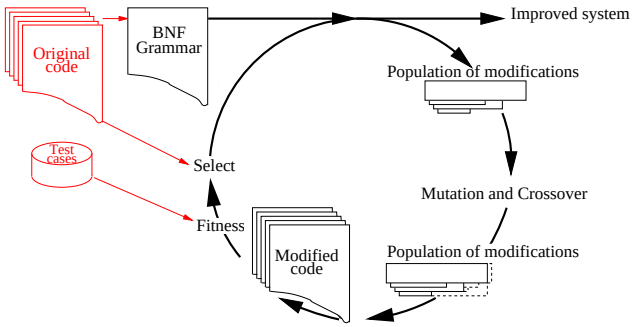

Figure 1: GP evolves patches to original C++ code.

type	Original Code	New Code
for2	`i < offsLenSampled`	`i < this->_nPat`
for2	`i < satup_->offs.size()`	`0`
for2	`j < satup_->offs.size()`	
	`vh = _mm_max_epu8(vh, vf);`	`vmax = vlo;`
	`pvFStore += 4;`	
	`_mm_store_si128(pvHStore,vh);`	`vh=_mm_max_epu8(vh,vf);`
	`ve = _mm_max_epu8(ve, vh);`	

Figure 2: Seven line change makes Bowtie2 77 times faster on targeted short DNA sequences.

After 25 hours (200 gens) GP's bloated answer was trimmed. This reduced the best from 39 changes to just 7 (Fig. 2).

DNA sequences from people not used in training the GP but from the same scanner were randomly selected. The evolved version took 3.9 hours. The released code took 12.2 days. In 89% of cases the GP version of Bowtie2 produced identical results. In 18 cases (9%) the GP version was better (i.e., the matches it reported had a mean Smith-Waterman score better than that of the released code). In one case the Smith-Waterman score was identical and in three (1.5%) the scores were worse but differed only in the 4^{th} and 6^{th} significant decimal place The median improvement was 0.1.

GISMOE has been used to optimise: legacy CUDA code for modern graphics hardware (EuroGP-2014 code also online), 3D NMR brain scan images (GECCO 2014), and MiniSAT (EuroGP-2104). See also Gen-O-Fix, and GenProg.

Reference

Langdon, W. B., and Harman, M. Optimising existing software with genetic programming. *IEEE Trans. on EC*.