IEEE Software

Tuesday, February 19, 2019

Software Improvement by Data Improvement

By: William B. Langdon, CREST Department of Computer Science, University College, London

Associate Editor: Federica Sarro, University College London (@f_sarro)

In previous blogs [1, 2] we summarised genetic improvement [3] and described the result of applying it to BarraCUDA [4]. Rather than giving a comprehensive update (see [5]) I describe a new twist: evolving software via constants buried within it to give better results. The next section describes updating 50000 free energy parameters used by dynamic programming to find the lowest energy state of RNA molecules and hence predict their secondary structure (i.e. how they fold) by fitting data in silico to known true structures. The last section describes converting a GNU C library square root function into a cube root function by data changes.

Better RNA structure prediction via data changes only

RNAfold is approximately 7 000 lines of code within the open source Vienna- RNA package. Almost all the constants within the C source code are pro- vided via 21 multi (1–6) dimensional int arrays [6, Tab. 2]. We used a population of 2000 variable length lists of operators to mutate these inte- gers. The problem dependent operators can invert values, replace them or update them with near by values. They can be applied to individuals values or using wild cards (*) sub-slices or even the whole of arrays. From these a population of mutated RNAfold is created. Each member of the popula- tion is tested on a 681 small RNA molecules and the mutants prediction is compared with their known structure [6, Tab. 1]. At the end of each gen- eration the members of the population are sorted by their average fitness on the 681 training examples and the top 1000 are selected to be parents of the next generation. Half the children are created by mutating one parent and the other 1000 by randomly combining two parents. After one hundred generations, the best mutant in the last generation is tidied (i.e. ineffective bloated parts of it are discarded) and used to give a new set of 50 000 integer parameters (29% of them are changed).

On average, on both big and small molecules of known structure (not used in training), the new version of RNAfold does better than the original. (In many cases it gives the same prediction, in some it is worse but in more it is better.)

Figure 1 shows RNAfold's original prediction of the secondary structure of an example RNA molecule and then the new prediction using the updated free energy parameters.

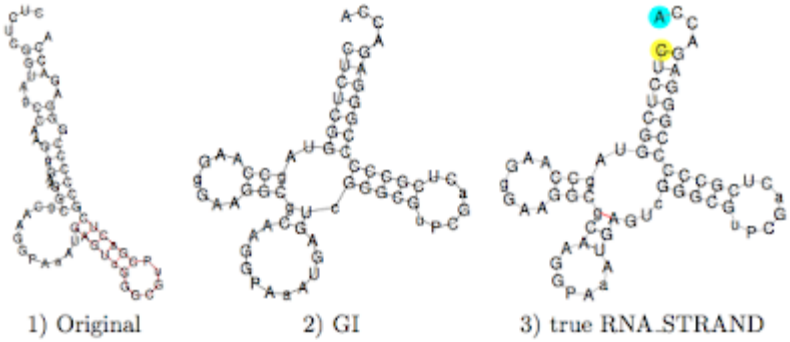

Figure 1: Secondary structure (i.e. folding patterns) for RNA molecule PDB 01001. 1) Prediction made original RNAfold does not match well true structure right. For example the highlighted hairpin loop (red) is not in the true structure. 2) Prediction made with GI parameters is almost identical to the true structure. 3) True structure. (Figure 2 tries to show the three dimensional structure of two PDB 01001 RNA molecules in a larger complex.)

A new Cube Root Function

The GNU C library contains more than a million constants. Most of these are related to internationalisation and non-ascii character sets [8]. However one implementation of the double precision square root function uses a table of 512 pairs of real numbers. (Most implementations of sqrt(x) simply call low level machine specific routines.) The table driven implementation is written in C and essentially uses three iterations of Newton-Raphson's method. To guarantee to converge on the correct square(x) to double precision accuracy, Newton-Raphson is given a very good start point for both the target value x^(1/2) and the derivative 0.5x^(−1/2) and these are held as pairs in the table.

Unlike the much larger RNAfold (previous section), with cbrt(x) some code changes were made by hand. These were to deal with: x being negative, normalising x to lie in the range 1.0 to 2, reversing the normalisation so that the answer has the right exponent and replacing the Newton-Raphson constant 1/2 by 1/3 [8, Sec. 2.1]. Given a suitable objective function (how close 23 is cbrt(x)×cbrt(x)×cbrt(x) to x), starting with each of the pairs of real numbers for sqrt(x), in less than five minutes CMA-ES [9] could evolve all 512 pairs of values for the cube root function.

The GNU C library contains many math functions which follow similar implementations. For fun, we used the same template to generate the log2(x) function [10].

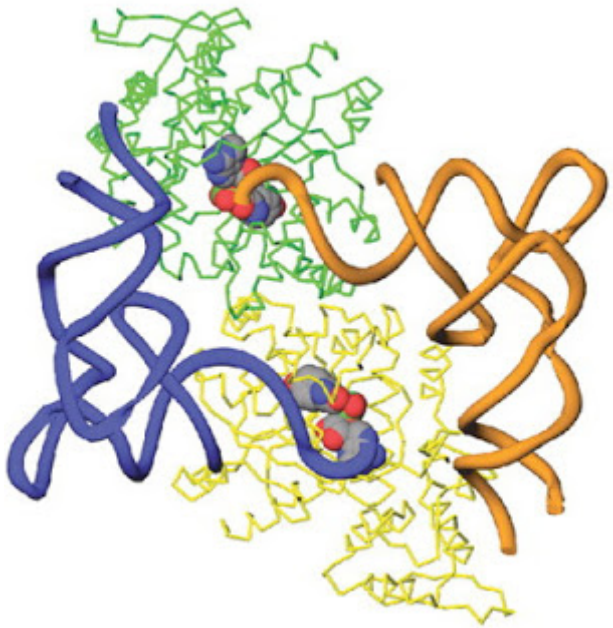

Figure 2: Three dimensional structure of two PDB 01001 RNA molecules (blue, orange) in a Yeast protein complex (green, yellow) [7, Fig 2. A].

References

1. W. B. Langdon and Justyna Petke. Genetic improvement. IEEE Soft- ware Blog, February 3 2016.
2. W. B. Langdon and Bob Davidson. BarraCUDA in the cloud. IEEE Software Blog, 8 January 2017.
3. W. B. Langdon and Mark Harman. Optimising existing software with genetic programming. IEEE Transactions on Evolutionary Computa- tion, 19(1):118–135, 2015.
4. W. B. Langdon and Brian Yee Hong Lam. Genetically improved Barra- CUDA. BioData Mining, 20(28), 2 August 2017.
5. Justyna Petke et al. Genetic improvement of software: a comprehensive survey. IEEE Transactions on Evolutionary Computation, 22(3):415– 432, 2018.
6. W. B. Langdon, Justyna Petke, and Ronny Lorenz. Evolving better RNAfold structure prediction. In Mauro Castelli et al., editors, EuroGP 2018, pages 220–236, Parma, Italy, 2018. Springer Verlag.
7. Masaru Tsunoda et al. Structural basis for recognition of cognate tRNA by tyrosyl-tRNA synthetase from three kingdoms. Nucleic Acids Re- search, 35(13):4289–4300, 2007.
8. W. B. Langdon and Justyna Petke. Evolving better software parame- ters. In Thelma Elita Colanzi and Phil McMinn, editors, SSBSE 2018, pages 363–369, Montpellier, France, 2018. Springer.
9. Nikolaus Hansen and Andreas Ostermeier. Completely derandom- ized self-adaptation in evolution strategies. Evolutionary Computation, 9(2):159–195, 2001.
10. W. B. Langdon. Evolving square root into binary logarithm. Technical Report RN/18/05, University College, London, UK, 2018.

You might also enjoy reading

- James Temperton. Code 'transplant' could revolutionise program- ming. Wired.co.uk, 30 July 2015. Online.
- John R. Woodward, Justyna Petke, and William Langdon. How com- puters are learning to make human software work more efficiently. The Conversation, page 10.08am BST, June 25 2015.
- Justyna Petke. Revolutionising the process of software development. DAASE project blog, August 7 2015.

Posted by Federica Sarro at 1:16 AM

No comments:

Post a Comment