Genetic Improvement of Computational Biology Software

William B. Langdon[⋆] and Karina Zile[⋆†]
[⋆]Department of Computer Science, †London Interdisciplinary Biosciences Consortium,
University College London, Gower Street, WC1E 6BT, UK.

ABSTRACT

There is a cultural divide between computer scientists and biologists that needs to be addressed. The two disciplines used to be quite unrelated but many new research areas have arisen from their synergy. We selectively review two multi-disciplinary problems: dealing with contamination in sequencing data repositories and improving software using biology inspired evolutionary computing. Through several examples, we show that ideas from biology may result in optimised code and provide surprising improvements that overcome challenges in speed and quality trade-offs. On the other hand, development of computational methods is essential for maintaining contamination free databases. Computer scientists and biologists must always be sceptical of each others data, just as they would be of their own.

Keywords genetic programming, GP, genetic improvement, GI, GGGP, search based software engineering, SBSE, software engineering, bioinformatics, next generation sequencing, NGS, DNA sequences, microarray, genechip, NCBI GEO, molecular biology, data cleansing, *in silico* contamination, identification and correction of mislabelled genes, big data cleanup, hitchhiking genes, 1k geneomes, 1KGP

ACM Reference format:
William B. Langdon and Karina Zile. 2017. Genetic Improvement of Computational Biology Software. In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017,* 4 pages.
DOI: http://dx.doi.org/10.1145/3067695.3082540

1 INTRODUCTION

Modern biology is in the middle of a civil war. On one side are those who might be thought of as traditionalists. Those who want biology to be the study of living organisms, preferably brightly coloured telegenic animals. These scientists want to catalogue the differences between every living thing. They want computer scientists to index their catalogues. Then there are other scientists who regard every living thing as the product of its genes. These tend to be microbiologists and can be classified as looking for the similarities between microbes and men. They want computer scientists to index their sequences.

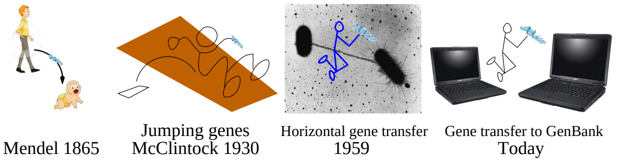

Figure 1: Tetratych showing: 1865 Mendel's [4] discovery of the essential digital nature of inheritance; 1930 Barbara McClintock's [5] discovery of transposons in Maize whereby genes move not only from parent to child but also along chromosomes; 1959 Micrograph of genetic transfer along a pilus linking two bacteria (Akiba and Ochia discovered the first interspecies gene transfer [6]); mycoplasma bacteria genes are transferred between computers, including into the reference human genome DNA sequence held by GenBank [2]. From "*In Silico* Infection of the Human Genome" [7, p245]

For the sake of simplicity we can choose the indexing of the reference human genome in 2000 [1] (see Section 2) as the start of a micro-biology data avalanche. The last seventeen years have seen an exponential growth in volume and processing rates of sequence data, with corresponding falls in costs, which even outstrip those that Moore's law has provided to the software industry. Perhaps computational biology provides a way for computer scientists to make a scientific contribution in their own right.

The next section reviews such a contribution: the discovery of *in silico* contamination of the reference human genome [2]. Then Section 3 gives a selective review of a more search based software engineering contribution: the use of evolutionary computation to optimise a few bioinformatics tools. Section 4 discusses the wider interaction between biology and computer science and the beneficial symbiosis that arises from it. In Section 5 we conclude that computer scientists still need to keep their wits about them, even when dealing with other people's data and that evolutionary computing, particularly genetic improvement (GI) [3], can play an important role in adapting tools in non-obvious Pareto trade-off optimisations.

2 DISBELIEVING: IS IT HUMAN OR IS IT A BUG?

Figure 1 reviews how our understanding of genes has changed in the last 150 years. The last of the four panels, represents the transmission of DNA sequences, usually as plain ascii text, across the globe. For example just before Christmas 2005 a

GECCO '17 Companion, Berlin, Germany

DOI: http://dx.doi.org/10.1145/3067695.3082540

bacteria sequence was accidentally upload along with thousands of human DNA sequences into the Data Bank of Japan. DDBJ and the National Center for Biotechnology Information shadow each other's data. So approximately 24 hours later, the bacteria sequence, now labelled Homo sapiens, was transferred to Washington, DC. Again NCBI and the European Bioinformatics Institute shadow their databases, so approximately 24 hours later the non-human sequence has made its way to the EBI in Cambridge. Notice in approximately 48 hours the bacteria gene has got itself transferred across the globe. Despite being reported eight years ago, it is still there.

Individual efforts to combat these "infections" resulted in software like SATIVA [8]. SATIVA is a tool to identify taxonomically mislabeled sequences given a multiple sequence alignment (MSA), a species tree and an evolutionary model. SATIVA performs well for genes with homologs across many species, but it is not suitable for analysing sequences with extremely fast or extremely slow divergence rates or imperfect data like chimeric or poor quality sequences. Currently no methods exist for identifying mislabelled sequences with no known homologs. Hence, there is a need for both new methods for automatic identification of mislabelled sequences and methods to remove identified mislabelled sequences from databases as quickly as they spread in the first place.

One DNA gene sequence did not stop in the NCBI databases (or their clones) but managed to get itself copied into physical devices. It lies in thousands of GeneChips manufactured by Affymetrix. It was via this route that we found it [2]. GeneChips give a bright response for each gene sequence coded into them when exposed to samples where the gene is expressed. Thus our hitch-hiking gene now gives itself away, when exposed to its own gene transcript. That is, if the sample is contaminated with the same bacteria, the GeneChip lights up in an unexpected way. Years later we were able to use this anomalous sequence to discover that about 1% of published gene expression data were contaminated with the bacteria (Mycoplasma) and so useless [7].

Although unfortunately since discontinued, the University of Essex's RNAnet [9] gave ready access to cleaned up and normalised data from tens of thousands of GeneChips [10] archived within NCBI's Gene Expression Omnibus GEO [11]. Nonetheless we have the cross correlation coefficients between all human genes across the many different tissues and disease states held by the GEO. These data are archived but are available should you have the wish and ability to deal with $22\,000 \times 22\,000$ arrays.

The Thousand Genomes Project [1] was a well funded flagship project of modern microbiology and yet we were able to discover [12] that more than 7% of their on-line data was from samples contaminated with Mycoplasma.

Nor is the contamination one way. Mark Longo et al. [13] found human DNA sequences appearing in 492 public databases which were supposed to contain DNA from species ranging from bacteria to plants and fish. Here again we suspect sloppy practice in microbiology laboratories leading to contamination of physical samples which were then sequences *en masse* causing the contaminating species (in this case Homo sapiens, rather than Mycoplasma) to be sequenced together with the target organism, and both sets of sequences being uploaded into a public data bank.

The NCBI holds more than 5 petabytes of on-line data. Cleaning it up this Augean mess will be a Herculean task.

3 OPTIMISING IS NOT JUST BEING FAST

Next we return to the traditional role of software engineering: making better programs. We selectively review some examples of using evolutionary computation, a biology inspired technique, not just to make existing programs faster, but also as a relatively automated way of exploring different trade-offs, particularly between speed and quality, and also of exploiting parallel hardware (see also [14]).

The first example is Bowtie2 [15] which is a state-of-the-art C++ program to align next generation short noisy DNA sequences against a reference genome. Although originally looking for a trade-off between speed and quality, evolutionary computation was able to find changes to the code which gave, for tasks like the one it had been trained on, spectacular speed ups with no loss of quality [16].

BarraCUDA [17] is another DNA alignment program, however it was written in C++/CUDA to run on nVidia GPU parallel hardware. Nevertheless a combination of manual effort and evolution (known as grow and graft genetic programming) was able to increase its performance by up to three fold [18]. The genetically improved (GIed) version of BarraCUDA was adopted and has been downloaded from SourceForge[1] many thousands of times. Indeed it has recently been demonstrated on epigenetic sequences supplied by Cambridge Epigenetix.

pknotsRG [19] was also a C program for which a CUDA version had been released. It works with RNA rather than DNA and predicts the folding pattern of short RNA molecules. Again using the GGGP approach, evolutionary computation was able to find small changes to manually written code which lead to spectacular speed ups (see Figure 2). For short RNA sequences the new code was up to ten thousand times faster [20].

More recent work [21] on another RNA prediction tool, RNAfold [22], has considered parallel vector SSE operations available within many modern CPUs rather than GPUs.

4 DISCUSSION

Our main focus is that biology (e.g. sequencing techniques, amount of data being generated) influences computer science (it has to be able to deal with all this data and accommodate limitations, contaminations, etc. in a meaningful way). While in turn changes in computer science (more scalable software, etc.) influences biology (e.g. by making previously prohibitively expensive computations possible and by developing contamination detection and containment tools). Changes in computer science are themselves inspired by biology. For example, genetic algorithms and neural networks. Biological

[1] http://sourceforge.net/projects/seqbarracuda/

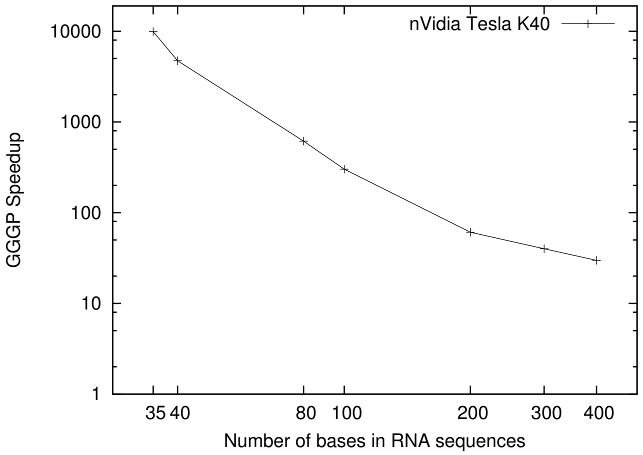

Figure 2: Ratio between original speed of CUDA version of pknotsRG and CUDA version after grow and graft change to allow processing multiple sequences in parallel for different RNA lengths. Note log scales.

inspiration has contributed greatly to new software. Similarly, changes in biology are also sometimes inspired by computer science. For example, thinking about genes in terms of character sequences and looking for similarity or divergence time between species in terms of string comparisons. Ideas from computer science give rise to new questions and to answer them more data is generated.

5 CONCLUSIONS

The oft heard remark that "our data *must* be good because we are using the data that the biologist use" roughly translates to "before I joined the project someone downloaded something from the Internet a previous version of which might have been cited by someone who might have worked once with a biologist".

There is a cultural disconnect. Computer scientists tend to think biological data must be ok, whereas microbiologists know that their laboratories are at risk of infection and cross contamination. However biologists tend not to even consider the possibility that their computers might hold contaminated data.

Yet in Section 2 we have seen many publically funded databases have a variety of new types of contamination. The big data volumes are just too huge for human curators and so there is a crying need for research to find new and better ways to automatically discover and remove suspect *in silico* contamination.

Being aware is the first step to avoid data contamination and misinterpretating computational results.

In Section 3 we reviewed recent work in which evolutionary computation is applied directly to several widely used computational biology software tools. Perhaps the most successful of these are the changes made using genetic improvement [23] to BarraCUDA. They have been adopted by the BarraCUDA team. Since integration of the GI evolved version, the team has made eight releases of BarraCUDA. These include fixing pre-GI bugs and supporting new GPU parallel computing devices, e.g. for use with epigenetic sequences [24].

Acknowledgements

I am grateful for the assistance of the anonymous reviewer. Teslas donated by nVidia.

REFERENCES

[1] Durbin, R.M., et al.: A map of human genome variation from population-scale sequencing. Nature **467**(7319) (2010) 1061–1073

[2] Aldecoa-Otalora, E., Langdon, W.B., Cunningham, P., Arno, M.J.: Unexpected presence of mycoplasma probes on human microarrays. BioTechniques **47**(6) (2009) 1013–1016

[3] Langdon, W.B.: Genetically improved software. In Gandomi, A.H., et al., eds.: Handbook of Genetic Programming Applications. Springer (2015) 181–220

[4] Mendel, G.: Experiments in plant hybrizization. Verhandlungen des naturforschenden Vereines in Brno (IV) (1865) 3–47 Translated by William Bateson in 1901 (updated Roger Blumberg, etc.).

[5] McClintock, B.: A cytological and genetical study of triploid maize. Genetics **14**(2) (1929) 180–222

[6] Akiba, T., Koyama, K., Ishiki, Y., Kimura, S., Fukushima, T.: On the mechanism of the development of multiple-drug-resistant clones of shigella. Japanese Journal of Microbiology **4** (1960) 219–227

[7] Langdon, W.B., Arno, M.: *In Silico* infection of the human genome. In Giacobini, M., et al., eds.: 10th European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics, EvoBIO 2012. Volume 7246 of LNCS., Malaga, Spain, Springer Verlag (2012) 245–249

[8] Kozlov, A.M., Zhang, J., Yilmaz, P., Gloeckner, F.O., Stamatakis, A.: Phylogeny-aware identification and correction of taxonomically mislabeled sequences. Nucleic Acids Research **44**(11) (2016) 5022–5033

[9] Langdon, W.B., Sanchez Graillet, O., Harrison, A.P.: RNAnet a map of human gene expression. arXiv:1001.4263 (2010)

[10] Langdon, W.B., Upton, G.J.G., da Silva Camargo, R., Harrison, A.P.: A survey of spatial defects in Homo Sapiens Affymetrix GeneChips. IEEE/ACM Transactions on Computational Biology and Bioinformatics **7**(4) (2009) 647–653

[11] Barrett, T., Troup, D.B., Wilhite, S.E., Ledoux, P., Rudnev, D., Evangelista, C., Kim, I.F., Soboleva, A., Tomashevsky, M., Edgar, R.: NCBI GEO: mining tens of millions of expression profiles–database and tools update. Nucleic Acids Research **35**(Database issue) (2007) D760–D765

[12] Langdon, W.B.: Mycoplasma contamination in the 1000 genomes project. BioData Mining **7**(3) (2014)

[13] Longo, M.S., O'Neill, M.J., O'Neill, R.J.: Abundant human DNA contamination identified in non-primate genome databases. PLoS ONE **6**(2) (2011) e16410

[14] Langdon, W.B., Lam, B.Y.H., Modat, M., Petke, J., Harman, M.: Genetic improvement of GPU software. Genetic Programming and Evolvable Machines **18**(1) (2017) 5–44

[15] Langmead, B., Salzberg, S.L.: Fast gapped-read alignment with Bowtie 2. Nature Methods **9**(4) (2012) 357–359

[16] Langdon, W.B., Harman, M.: Optimising existing software with genetic programming. IEEE Transactions on Evolutionary Computation **19**(1) (2015) 118–135

[17] Klus, P., Lam, S., Lyberg, D., Cheung, M.S., Pullan, G., McFarlane, I., Yeo, G.S.H., Lam, B.Y.H.: BarraCUDA - a fast short read sequence aligner using graphics processing units. BMC Research Notes **5**(27) (2012)

[18] Langdon, W.B., Lam, B.Y.H., Petke, J., Harman, M.: Improving CUDA DNA analysis software with genetic programming. In Silva, S., et al., eds.: GECCO, Madrid, ACM (2015) 1063–1070

[19] Reeder, J., Steffen, P., Giegerich, R.: pknotsRG: RNA pseudoknot folding including near-optimal structures and sliding windows. Nucleic Acids Research **35**(suppl 2) (2007) W320–W324

[20] Langdon, W.B., Harman, M.: Grow and graft a better CUDA pknotsRG for RNA pseudoknot free energy calculation. In Langdon, W.B., et al., eds.: Genetic Improvement 2015 Workshop, Madrid, ACM (2015) 805–810

[21] Langdon, W.B., Lorenz, R.: Improving SSE parallel code with grow and graft genetic programming. In Petke, J., et al., eds.: GI-2017, Berlin (2017)

[22] Lorenz, R., Bernhart, S.H., Höner zu Siederdissen, C., Tafer, H., Flamm, C., Stadler, P.F., Hofacker, I.L.: ViennaRNA package 2.0. Algorithms for Molecular Biology **6**(1) (2011)

[23] Petke, J., Haraldsson, S.O., Harman, M., Langdon, W.B., White, D.R., Woodward, J.R.: Genetic improvement of software: a comprehensive survey. (IEEE Transactions on Evolutionary Computation) In press.

[24] Langdon, W.B., Vilella, A., Lam, B.Y.H., Petke, J., Harman, M.: Benchmarking genetically improved BarraCUDA on epigenetic methylation NGS datasets and nVidia GPUs. In Petke, J., et al., eds.: Genetic Improvement 2016 Workshop, Denver, ACM (2016) 1131–1132