Bioinformatics Advance Access published November 28, 2015

Subject Section

Comment on: “ERGC: An efficient referential genome compression algorithm”

Sebastian Deorowicz [1,*], Szymon Grabowski [2], Idoia Ochoa [3*], Mikel Hernaez [3], and Tsachy Weissman [3]

[1] Institute of Informatics, Silesian University of Technology, Akademicka 16, Gliwice, 44-100, Akademicka 16, Poland,
[2] Institute of Applied Computer Science, Lodz University of Technology, Al. Politechniki 11, 90-924 Łódź,
[3] Department of Electrical Engineering, Stanford University, 350 Serra Mall, Stanford, CA.

* To whom correspondence should be addressed.

Associate Editor: Dr. John Hancock

Abstract

Motivation: Data compression is crucial in effective handling of genomic data. Among several recently published algorithms, ERGC seems to be surprisingly good, easily beating all of the competitors.
Results: We evaluated ERGC and the previously proposed algorithms GDC and iDoComp, which are the ones used in the original paper for comparison, on a wide data set including 12 assemblies of human genome (instead of only four of them in the original paper). ERGC wins only when one of the genomes (referential or target) contains mixed-cased letters (which is the case for only the two Korean genomes). In all other cases ERGC is on average an order of magnitude worse than GDC and iDoComp.
Contact: sebastian.deorowicz@polsl.pl, iochoa@stanford.edu
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

The rapid growth of genomic data in the last few years demands for efficient compression methods to facilitate their storage and transfer. One of the standard problems of this kind is (referential) compression of multiple individual genomes of the same species (Kuruppu *et al*., 2011; Deorowicz and Grabowski, 2011; Wandelt and Leser, 2013; Ochoa *et al*., 2014; Deorowicz *et al*., 2015). As a pair of individual genomes differ only in variant loci, we can expect radical savings in storing a referentially encoded genome.

Recently, Saha and Rajasekaran (2015) published a new referential genome compression algorithm, called ERGC. As the experimental results presented in this paper seemed to suggest that ERGC is surprisingly good, winning easily in compression ratio over the previously proposed algorithms GDC (Deorowicz and Grabowski, 2011) and iDoComp (Ochoa *et al*., 2014), we decided to take a closer look at ERGC and repeat the experiments on a wider set of data.

2 Discussion

In the first experiment we repeated the referential compression evaluation from the ERGC paper, where the $D_1, \ldots, D_5$ experiments are specified in Table 1 in the cited paper and the actual results are shown there in Table 2. In each D_i experiment all 24 chromosomes of one human genome are referentially compressed, given some other human genome as a reference. We repeated these experiments in our Table 1.

As it can be observed, the results differ significantly for cases D_1 and D_5. First, the cited paper incorrectly reported 'NA' for the GDC algorithm in these two cases. Moreover, iDoComp achieves 6,288 KB on D_1 and 33,158 KB on D_5, rather than 65,708.47 KB and 209,380.79 KB as reported in the cited paper (note the order of magnitude difference). Finally, applying ERGC on D_5 resulted in a compressed file of 9,373 KB, instead of the reported 19,396.40 KB.

With the corrected results, we concluded that ERGC achieves the best compression ratio in all cases except in D_1, where it is outperformed by both GDC and iDoComp. Taking a closer look, we observed that in all but D_1 one of the KOREAN genomes is used either as a reference or a target genome. The KOREAN genomes differ from the other ones in that they contain both lower and upper case letters (see Supplementary material). They also contain non-standard symbols, similarly to the YH genome.

Table 3. Summary of the results for all the considered pairs both for chromosomes 10 and 20.

	Chromosome 10				Chromosome 20			
Reference	Target size	GDC size	iDoComp size	ERGC size	Target size	GDC size	iDoComp size	ERGC size
CHM1_1.0	1,495,816	10,229	11,805	289,929	694,646	4,242	5,132	135,412‡
CHM1_1.1	1,496,019	9,505	11,054	290,014	694,641	4,252	5,088	100,306
CSA	1,497,653	51,057	59,500†	339,998	695,024	13,741	15,695	146,233
HG17	1,496,426	8,146	7,801	167,270	695,127	3,457	3,893	48,193
HG18	1,496,465	8,106	7,753	145,209	695,127	3,457	3,893	48,193
HG19	1,496,303	8,153	7,913	227,746	694,529	3,472	3,919	48,294
HG38	1,498,065	9,329	9,543	222,986	693,090	2,743	2,798	78,216
HuRef	1,502,946	12,680	15,582	295,338	697,946	5,037	5,862	95,444
KO131	1,496,143	16,388	18,246†	158,849	694,978	6,169	6,984†	75,827
KO224	1,496,143	14,736	16,997†	158,635	694,978	5,486	6,543†	68,173
WGSA	1,503,392	25,607	31,897	336,064	697,939	10,403	12,470	137,623
YH	1,496,143	9,388	8,785	153,532	694,978	4,063	4,406	49,951

‡This total ERGC result is not certain, as ERGC crashed on the pair of sequences: CHM1_1.0 (reference), CHM1_1.1 (target). †These total iDoComp results are taken from its bug-fix version (`https://github.com/mikelhernaez/iDoComp`, v1.2) as the earlier version v1.1 crashed on 5 pairs of sequences.

Table 1. Re-run experiments from Table 2 in the ERGC paper.

Dataset	Target size	GDC size	iDoComp size	ERGC size
D_1	3,132	6,439	6,288	7,890
D_2	3,132	29,831	30,437	9,217
D_3	3,132	35,227	33,926	9,404
D_4	2,938	12,293	7,213	4,914
D_5	3,124	34,806	33,158	9,373

Table 2. Re-run experiments from Table 1 above, with the difference that the KOREAN genomes are converted to upper case.

Dataset	Target size	GDC size	iDoComp size	ERGC size
D_2	3,132	7,660	7,473	9,273
D_3	3,132	7,849	7,691	9,449
D_4	2,938	1,073	1,020	1,246
D_5	3,124	7,820	7,754	9,425

The D_1 experiment is skipped run since since both the reference and the target in it consist of upper case DNA symbols only.

To better understand if the potential of ERGC is due to the design of the algorithm itself, or to the handling of the lower and upper case letters and/or the non-standard symbols, we simulated these experiments again ($D_2, \ldots, D_5$) with the KOREAN genomes transformed to all upper case. Surprisingly, as demonstrated in Table 2, ERGC achieved the worst compressed size in all cases. The reported results suggest that the gain of ERGC comes from the handling of the lower and upper case symbols.

Since the previous experiments may not be representative enough, we decided to use a wider set of data. Specifically, we chose the human genome datasets from the GDC 2 paper (Deorowicz *et al.*, 2015), which is a superset of the collections from (Deorowicz and Grabowski, 2011) and (Ochoa *et al.*, 2014). To reduce the amount of computations, we chose only two chromosomes from each genome: chr10 and chr20. Then, to follow the ERGC paper methodology, we tested referential compression in pairs of chromosomes, in a round-robin fashion. In Table 3 we show the sums of compressed sizes; detailed results are presented in the Supplementary material. ERGC is outperformed in all cases by GDC and iDoComp.

3 Conclusion

From the conducted experiments we can draw several conclusions. Firstly, the reported compression results on the D_1 and D_5 cases in the cited paper are not correct for both GDC and iDoComp. Secondly, ERGC seems to get the main advantage due to specific handling of lower and upper case difference between the target and the reference genome. We believe that using the KOREAN genomes is fair, however, the authors should have provided simulations on a wider set of data where the gain does not come from the lower and upper case difference. Finally, after running several experiments on different datasets, we can conclude that ERGC performs in general poorly when compared with GDC and iDoComp.

Funding

This work has been supported by a fellowship from the Basque Government, a grant from the Center for Science of Information (CSoI), the 2014-07364-01 NIH grant and the 1157849-1-QAZCC NSF grant. It is also supported by the Polish National Science Centre under the project DEC-2011/03/B/ST6/01588. The infrastructure supported by POIG.02.03.01-24-099/13 grant: 'GeCONiI—Upper Silesian Center for Computational Science and Engineering'.

References

Deorowicz, S., Danek, A., Niemiec, M. (2015) GDC 2: Compression of large collections of genomes, *Scientific Reports*, **5(11565)**.

Deorowicz, S., Grabowski, S. (2011) Robust relative compression of genomes with random access, *Bioinformatics*, **27(21)**, 2979–2986.

Kuruppu, S., Puglisi, S., Zobel, J. (2011) Optimized relative Lempel-Ziv compression of genomes. In *Proceedings of the ACSC Australasian Computer Science Conference* Reynolds M. (ed.), Australian Computer Society, Inc., Sydney, Australia, pp. 91–98.

Ochoa, I., Hernaez, M., Weissman, T. (2015) iDoComp: a compression scheme for assembled genomes, *Bioinformatics*, **31(5)**, 626–633.

Saha, S., Rajasekaran, S. (2015) ERGC: An efficient referential genome compression algorithm, *Bioinformatics*, doi: 10.1093/bioinformatics/btv399.

Wandelt, S., Leser, U. (2013) FRESCO: Referential compression of highly similar sequences, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **10(5)**, 1275–1288.