The Carbon Footprint of Bioinformatics

Jason Grealey,*[,†,1,2] Loïc Lannelongue,[†,3,4,5] Woei-Yuh Saw,[1] Jonathan Marten,[‡,4] Guillaume Méric,[1,6] Sergio Ruiz-Carmona,[1] and Michael Inouye*[,1,3,4,5,7,8]

[1]Cambridge Baker Systems Genomics Initiative, Baker Heart and Diabetes Institute, Melbourne, VIC, Australia

[2]Department of Mathematics and Statistics, La Trobe University, Melbourne, VIC, Australia

[3]Cambridge Baker Systems Genomics Initiative, Department of Public Health and Primary Care, University of Cambridge, Cambridge, United Kingdom

[4]British Heart Foundation Cardiovascular Epidemiology Unit, Department of Public Health and Primary Care, University of Cambridge, Cambridge, United Kingdom

[5]Health Data Research UK Cambridge, Wellcome Genome Campus and University of Cambridge, Cambridge, United Kingdom

[6]Department of Infectious Diseases, Central Clinical School, Monash University, Melbourne, VIC, Australia

[7]British Heart Foundation Centre of Research Excellence, University of Cambridge, Cambridge, United Kingdom

[8]The Alan Turing Institute, London, United Kingdom

[†]These authors contributed equally to this work as first authors.

[‡]Present address: Genomics PLC, King Charles House, Park End Street, Oxford, United Kingdom

***Corresponding authors:** E-mails: mi336@medschl.cam.ac.uk, minouye@baker.edu.au; jason.grealey@baker.edu.au.

Associate editor: Sudhir Kumar

Abstract

Bioinformatic research relies on large-scale computational infrastructures which have a nonzero carbon footprint but so far, no study has quantified the environmental costs of bioinformatic tools and commonly run analyses. In this work, we estimate the carbon footprint of bioinformatics (in kilograms of CO_2 equivalent units, $kgCO_2e$) using the freely available Green Algorithms calculator (www.green-algorithms.org, last accessed 2022). We assessed 1) bioinformatic approaches in genome-wide association studies (GWAS), RNA sequencing, genome assembly, metagenomics, phylogenetics, and molecular simulations, as well as 2) computation strategies, such as parallelization, CPU (central processing unit) versus GPU (graphics processing unit), cloud versus local computing infrastructure, and geography. In particular, we found that biobank-scale GWAS emitted substantial $kgCO_2e$ and simple software upgrades could make it greener, for example, upgrading from BOLT-LMM v1 to v2.3 reduced carbon footprint by 73%. Moreover, switching from the average data center to a more efficient one can reduce carbon footprint by approximately 34%. Memory over-allocation can also be a substantial contributor to an algorithm's greenhouse gas emissions. The use of faster processors or greater parallelization reduces running time but can lead to greater carbon footprint. Finally, we provide guidance on how researchers can reduce power consumption and minimize $kgCO_2e$. Overall, this work elucidates the carbon footprint of common analyses in bioinformatics and provides solutions which empower a move toward greener research.

Key words: **carbon footprint, bioinformatics, genomics, green algorithms.**

Introduction

Biological and biomedical research now requires the analysis of large and complex data sets, which would not be possible without the use of large-scale computational resources. Although bioinformatic research has enabled major advances in the understanding of a myriad of diseases such as cancers (ICGC/TCGA Pan-Cancer Analysis of Whole Genomes Consortium 2020; Kachuri et al. 2020; PCAWG Structural Variation Working Group et al. 2020) and COVID-19 (The Severe Covid-19 GWAS Group 2020), the costs of the associated computing requirements are not limited to the financial; the energy usage of computers causes greenhouse gas (GHG) emissions which themself have a detrimental impact on human health.

Energy production affects both human and planetary health. The yearly electricity usage of data centers and high-performance computing facilities (200 TWh; Jones 2018) already exceeds the consumption of countries such as Ireland or Denmark (Primary Energy Consumption by World Region 2021) and is predicted to continue to rise over the next decade (Andrae and Edler 2015; Jones 2018). Power generation, through the associated emissions of GHGs, is one of the main causes of both outdoor air pollution and climate change. Every year, it is estimated that 4.2 million deaths are caused by ambient air pollution alone, whereas 91% of the world's population suffers from air quality below the World Health Organisation standards (Air Pollution 2016). Global warming

Article

results in further consequences on human health, economy, and society: the daily population exposure to wildfires has increased in 77% of countries (Watts et al. 2019), 133.6 billion potential work hours were lost to high temperatures in 2018 and with 220 million heatwave exposures, vulnerable populations (aged 65 years and older) are affected at an unprecedented level.

The growth of large biological databases, such as UK BioBank (Bycroft et al. 2018), All of Us Initiative (National Institutes of Health [NIH] – All of Us n.d.), and Our Future Health (Accelerating Detection of Disease – UK Research and Innovation n.d.), has substantially increased the need for computational resources to analyze these data and will continue to do so. With climate change an urgent global emergency, it is important to assess the carbon footprint of these analyses and their requisite computational tools so that environmental impacts can be minimized.

Other fields of science, such as machine learning (Strubell et al. 2019; Bender et al. 2021) and astrophysics (Jahnke et al. 2020; Portegies Zwart 2020; Stevens et al. 2020), have started to investigate the environmental impact of their computational work; this highlights the need for such study in computational biology. Notwithstanding that, alongside computation, various other aspects of biological research are responsible for substantial GHG emissions. For example, it has been estimated that powering the equipment of a typical (7–10 people) life sciences laboratory likely generates more than 20 metric tons of CO_2e annually (Nathans and Sterling 2016). Travel also contributes to science's carbon footprint, the carbon footprint of the annual meeting of the Society for Neuroscience (which has around 30,000 attendees) has been estimated to be approximately 22,000 metric tons CO_2e (Nathans and Sterling 2016), roughly equivalent to the annual carbon footprint of 1,000 medium sized laboratories.

In this study, we estimate the carbon footprint of common bioinformatic tools using a model which accounts for the energy use of different hardware components and the emissions associated with electricity production. Since metrics for carbon emissions are relatively unfamiliar to most scientists, we compare the results with distances traveled by car (an average European car emits 0.175 kgCO_2e/km; Greenhouse Gas Reporting: Conversion Factors 2019 n.d.; Helmers et al. 2019) and amounts of carbon sequestered by trees (a mature tree sequesters approximately 0.917 kgCO_2e per month; Lannelongue et al. 2021). This study raises awareness, provides easy-to-use metrics, and makes recommendations for greener bioinformatics.

Results

We estimated the carbon footprint of a variety of bioinformatic tools and analyses (table 1) using the Green Algorithms model and online tool (see Materials and Methods). For each software, we utilized benchmarks of running time and computational resources; in the rare cases where published benchmarks were unavailable, we used in-house analyses to estimate resource usage (see Materials and Methods). The results depend on the efficiency of the computing facility measured by its power usage effectiveness (PUE), which quantifies the additional energy the data center needs, for example, for cooling and lighting. The estimations here are based on the global average PUE of 1.67, that is, an extra 67% is necessary compared with what the servers alone demand. The global average carbon intensity (CI) (0.475 kgCO_2e/kWh; Emissions – Global Energy & CO2 Status Report 2019 – Analysis 2019) is also used and we assume processing cores (CPU or GPU) are fully used (usage factor of 1) (see Materials and Methods).

We considered a wide range of bioinformatic analyses: genome assembly, metagenomics, phylogenetics, RNA sequencing (RNAseq), genome-wide association analysis, molecular simulations, and virtual screening. We also show that choices of hardware substantially affect the carbon footprint of a given analysis, in particular cloud versus local computing platforms, memory usage, processor options, and parallel computing. The same applies to software choices, including software versions. These results present orders of magnitude and we note how the estimations are likely to scale with different parameters (e.g., sample size or number of features), but for precise estimations of specific analysis, scientists should estimate their own footprint, for example using the Green Algorithms tool (www.green-algorithms.org, last accessed 2022).

Genome Assembly

Genome assembly is the process of combining sequencing reads (short or long reads, or a combination) into a single or a set consensus sequences for an organism. Hunt et al. (2014) compared SSPACE (Boetzer et al. 2011), SGA (Simpson and Durbin 2012), and SOAPdenovo2 (Luo et al. 2012) for genome scaffolding using contigs produced with the Velvet assembler (Zerbino and Birney 2008) and the human chromosome 14 GAGE data set (Salzberg et al. 2012); two read sets were compared, one using 22.7 million short reads (fragment length of 3 kb) and the other 2.4 million long reads (35 kb). Scaffolding the short or long reads resulted in similarly low carbon footprints (0.0010 to 0.13 kgCO_2e) (table 1). However, SGA had a carbon footprint up to 49 times higher than the other tools (table 1), but it may be a result of the increased time needed to build the FM-index (full-text minute-space index) (Simpson and Durbin 2012). As the running time of many genome assembly tools scale linearly with the number of reads (Sutton et al. 2019), these results equate to between 0.00012 to 0.0057 kgCO_2e (0.00013 to 0.0063 tree-months) per million short reads assembled and 0.00043 to 0.012 kgCO_2e (0.00047 to 0.013 tree-months) per million long reads assembled. On an average, long read assembly had a carbon footprint per million reads 3.2x larger than short-read assembly for the tools we measured. All three methods had similar performance on these read sets with SOAPdenovo2 slightly outperforming SGA and SSPACE.

For whole genome assembly of humans, ABySS (Jackman et al. 2017) and MEGAHIT (Li et al. 2016) were benchmarked by Jackman et al. (2017) using Illumina short read sequencing (815 M reads, 379 M uniquely mapped reads, 6 kb mean

Table 1. Carbon Footprint of a Range of Bioinformatic Tasks.

Task	Tool	Version	Details about the Experiments	Carbon Footprint		Tree-months	km in a Car (EU)	Running Time and Memory	Approximate Scaling (if known)
				Increase (%)	$kgCO_2e$				
Genome scaffolding	SSPACE	2.0	Scaffolding 2.4 million long reads from human chromosome 14 (Hunt et al. 2014).	—	0.0010	0.0011	0.01	3 min 21 s 30 GB	Linearly with number of reads.
	SOAPdenovo2	r223		+45%	0.0015	0.0016	0.01	4 min 52 s 30 GB	
	SGA	0.9.43		+2,752%	0.029	0.032	0.17	1 h 35 min 30 GB	
Genome scaffolding	SSPACE	2.0	Scaffolding 23 million short reads from human chromosome 14 (Hunt et al. 2014).	—	0.0027	0.0029	0.02	8 min 40 s 30 GB	
	SOAPdenovo2	r223		+34%	0.0036	0.0039	0.02	1 min 38 s 30 GB	
	SGA	0.9.43		+4,801%	0.13	0.14	0.74	7 h 05 min 30 GB	
Genome assembly	Abyss	2.0	De novo assembly of a human genome from Illumina sequencing reads (Jackman et al. 2017).	—	11	12	61	20 h 34 GB	
	MEGAHIT	1.0.6		+42%	15	16	86	26 h 197 GB	
Metagenome assembly	MetaVelvet k101	1.2.01	Metagenome assembly from 100 soil samples (Vollmers et al. 2017).	—	14	16	82	1 h 06 min 130 GB	
	MEGAHIT	1.0.3		+438%	77	84	439	15 h 36 min 12 GB	
	metaSPAdes	3.8.0		+1,206%	186	203	1,065	29 h 24 min 60 GB	
Metagenome classification (short read)	Kraken2	2.0.7	Metagenomic classification of 5 Gb of randomly sampled reads from Zymo mock community (batch ZRC190633), containing yeast, Gram-negative, and positive bacteria (Dilthey et al. 2019)	—	0.0052	0.0057	0.03	20 min 21 GB	Linearly with number of reads.
	Centrifuge	1.0.4		+141%	0.013	0.014	0.07	58 min 12 GB	
	Kraken/Bracken	0.10.5/1.0.0		+1,650%	0.092	0.10	0.52	1 h 40 min 154 GB	
Metagenome classification (long read)	MetaMaps	—		—	18.25	19.91	104.27	209 h 53 min 262 GB	
Phylogenetics	BEAST/BEAGLE	1.8.4/2.1.2	Codon substitution modeling of extant carnivores and a pangolin group. Nucleotide substitution and phylogeographic modeling of Ebola virus genomes. See supplementary table 2, Supplementary Material online, for detailed results (Baele et al. 2019).	—	0.012–0.30	0.013–0.33	0.069–1.72	3 min 30 s to 7 h 45 min 2–8 GB	Power law with number of loci.

(continued)

Table 1. Continued

Task	Tool	Version	Details about the Experiments	Carbon Footprint		Tree-months	km in a Car (EU)	Running Time and Memory	Approximate Scaling (if known)
				Increase (%)	$kgCO_2e$				
Phylogenetics	RAxml/ExaML, PhyML, IQ-TREE, FastTree	8.2.0/3.0.17, 20160530 1.4.2, 2.1.9	Over 670,000 tree inferences on about 45,000 single-gene alignments and supermatrices from 19 empirical phylogenomic data sets with thousands of genes and around 200 taxa. (Zhou et al. 2018)	—	3565	3889	20,371	300,000 h 8 GB	
Phylogenetics	ExaML	—	A 322-million-bp MULTIZ alignment of putatively orthologous genome regions across all species, comprising approximately 30% of an average assembled avian genome. This corresponded to the maximal orthologous sequence obtainable across all orders of *Neoaves*.(Jarvis et al. 2014)	—	4372	4769	24,983	367,920 h 8 GB	
RNA read alignment	HISAT2	2.0.0beta	Alignment of 10 million 100-base read pairs to Homo Sapiens hg19 genome (Baruzzo et al. 2017).	—	0.0054	0.0059	0.031	1 min 48 s 5 GB	Linearly with number of reads.
	STAR	2.5.0a		+78%	0.0097	0.011	0.055	6 min 01 s 35 GB	
	TopHat2	2.1.0		+5,756%	0.32	0.35	1.81	2 h 14 min 16 GB	
	Novoalign	3.02.13		+17,926%	0.98	1.07	5.58	32 h 12 min 64 GB	
RNA read alignment	HISAT2	2.0.0beta	Alignment of 10 million 100-base read pairs to Plasmodium falciparum genome (Baruzzo et al. 2017).	—	0.0052	0.0057	0.030	1 min 44 s 1 GB	
	TopHat2	2.1.0		+4,519%	0.24	0.26	1.37	1 h 25 min 13 GB	
	STAR	2.5.0a		+7,025%	0.37	0.40	2.11	2 h 27 min 8 GB	
	Novoalign	3.02.13		+12,847%	0.67	0.73	3.83	38 h 04 min 21 GB	
RNA-seq QC pipeline	FastQC, TrimGalore, bbmap/clumpify, and STAR	-/v0.6.0/-/v2.7.0e	Quality control analysis of raw reads quality of 392 samples from the Childhood Asthma Study (in-house).	—	54.97	59.97	314.11	485 h 12 min 8 GB	
Transcript isoform abundance estimation	Sailfish 1 core	0.6.3	Transcript isoform quantification of 100 million in silico reads generated from Flux Simulator with hg19 genome	—	0.0081	0.0088	0.046	42 min 7 GB	Linearly with the number of reads.
	Sailfish 16 cores			+344%	0.036	0.039	0.21	14 min 7 GB	

(continued)

Table 1. Continued

Task	Tool	Version	Details about the Experiments	Carbon Footprint		Tree-months	km in a Car (EU)	Running Time and Memory	Approximate Scaling (if known)
				Increase (%)	$kgCO_2e$				
	Cufflinks 1 core	2.1.1	and GENCODE v19 annotation set (Kanitz et al. 2015)	+451%	0.045	0.049	0.26	3 h 30 min 11 GB	
	Cufflinks 16 cores			+3,262%	0.27	0.30	1.56	1 h 45 min 12 GB	
	RSEM 1 core	1.2.18		+6,982%	0.57	0.63	3.28	47 h 10 min 9 GB	
	RSEM 16 cores			+17,162%	1.40	1.53	8.00	8 h 50 min 21 GB	
GWAS	Bolt-LMM	2.3	Analyses of a single trait in UK Biobank ($N = 500{,}000$) (Loh et al. 2018)	—	4.70	5.13	26.87	60 h 58 min 100 GB	Linearly with number of variants.
	Bolt-LMM	1.0		+268%	17.29	18.86	98.81	224 h 10 min 100 GB	
Cohort scale eQTL analysis	TensorQTL	1.0.2	Cis-eQTL mapping of 10.7 M SNPs against 18,373 genetic features in a cohort of 2,745 individuals (in-house).	—	2.04	2.22	11.7	1 h 14 min 192 GB	Nonlinearly with the number of traits or the sample size.
	LIMIX	2.0.3		+9,256%	190.73	208.07	1,089.9	9,705 h 41–221 GB	
Single cis-eQTL gene mapping	TensorQTL	—	Cis-eQTL mapping one gene from skeletal muscle in GTEx (v6p) (Taylor-Weiner et al. 2019).	—	0.00001	0.00001	0.00004	0.11 s 52 GB	
	FastQTL	—		+2,681%	0.0002	0.0002	0.001	30 s 52 GB	
Molecular dynamics simulation	AMBER	18	Simulation of a Satellite Tobacco Mosaic Virus with 1,066,628 atoms for 100 ns[a] (NAMD Performance n.d.; The Pmemd.Cuda GPU Implementation n.d.).	—	18	19	102	75 h ([b])	
	NAMD	2.13		+433%	95	104	544	400 h ([b])	
Molecular docking	Glide	57111	Molecular docking of four DUD systems, scaled to 1 m ligands (Ruiz-Carmona et al. 2014)	—	13	14	74	1,027 h 47 min 0.05 GB	
	rDock	—		+1,092%	154	168	878	12,250 h 0.05 GB	
	AutoDock Vina	—		+3,886%	514	561	2,938	40,972 h 0.05 GB	

NOTE.—Further details for each task are included in supplementary additional file 1, Supplementary Material online.

[a]Note different simulation parameters between the two: AMBER18 (4fs timestep, 9 A cut-off) NAMD (2fs timestep with rigid bonds, 12 A cut-off with PME every two steps).

[b]No memory included due to a lack of information.

insert size) (table 1). We estimated the carbon footprint of these tasks to be between 11 and 15 $kgCO_2e$ (12 to 16 tree-months), or per million reads, between 0.013 and 0.019 $kgCO_2e$ (0.014–0.020 tree-months). It is difficult to succinctly quantify the accuracy of these tools as it has been shown to vary greatly between use cases and data sets (Bradnam et al. 2013). Instead, relevant published benchmarks, such as Bradnam et al. (2013), Lischer and Shimizu (2017), and Jackman et al. (2017) can indicate the assembler that excels in the area of interest, for example, number of error-free bases, coverage, or continuity.

Metagenomics

Metagenomics is the sequencing and analysis of all genetic material in a sample. Based on a benchmark by Vollmers et al. (2017), we estimated the carbon footprint of metagenome assembly with three commonly used assemblers, metaSPAdes (Nurk et al. 2017), MEGAHIT (Li et al. 2016), and MetaVelvet (k-mer length 101 bp) (Namiki et al. 2012) on 100 samples from forest soil (33 M reads, median length 360 bp). It ranged between 14 and 186 $kgCO_2e$ (table 1), corresponding to 0.14 to 1.9 $kgCO_2e$ per sample (0.2–2 tree-months). MetaSPAdes had the greatest carbon footprint but also the best performance followed by MetaVelvet and MEGAHIT, respectively.

For metagenomic classifiers, Dilthey et al. (2019) benchmarked MetaMaps (Dilthey et al. 2019), Kraken2 (Wood et al. 2019), Kraken/Bracken (Wood and Salzberg 2014; Lu et al. 2017), and Centrifuge (Kim et al. 2016). They compared these tools on approximately 5 Gb of randomly sampled reads from an Oxford Nanopore GridION sequencing run from Zymo mock communities, which comprises five Gram-positive bacteria, three Gram-negative bacteria and two types of yeast. Carbon footprints differed by several orders of magnitude, 18.25 $kgCO_2e$ for the long-read classifier MetaMaps but less than 0.1 $kgCO_2e$ for the short-read classifiers (table 1). The carbon footprints per Gb of classified reads ranged from 0.001 to 0.018 $kgCO_2e$ (0.001 to 0.02 tree-months) using the short-read classifiers (Kraken2, Centrifuge, Kraken/Bracken) and 3.65 $kgCO_2e$ (4 tree-months) when using MetaMaps. Kraken2 had the highest performance over all taxonomic ranks when all reads were assembled, followed by Kraken/Bracken, Centrifuge, and MetaMaps. However, when considering long reads (>1,000 bp), MetaMaps had the highest precision and recall for all available taxonomic levels, followed by Kraken2, Kraken/Bracken, and Centrifuge.

Phylogenetics

Phylogenetics is the use of genetic information to analyze the evolutionary history and relationships among individuals or groups. Baele et al. (2019) benchmarked nucleotide substitution models with and without spatial location information to study the evolution of the Ebola virus during the 2013–2016 West African epidemics (1,610 genomes, 18,992 nucleotides; Dudas et al. 2017). These nucleotide substitution models are based on a four-partition model (one for each codon position and one for the intergenic region), and generalized linear models (Dudas et al. 2017) when including spatial information in the phylogeographic analysis. Additionally, Baele et al. benchmarked more complex Goldman and Yang's (1994) codon substitution models on a set of mitochondrial genome from extant carnivores and a pangolin outgroup. For all these tasks, they utilized the Bayesian inference framework implemented in BEAST (Drummond et al. 2012) combined with BEAGLE (Ayres et al. 2012) for computational speedup.

We estimated the carbon footprint of nucleotide-based modeling of the Ebola virus data set was between 0.012 and 0.076 $kgCO_2e$ depending on hardware choices and up to 25 times higher (up to 0.30 $kgCO_2e$) when including spatial information. More complex codon modeling of extant carnivores and pangolins resulted in a greater footprint, from 0.017 to 0.10 $kgCO_2e$ (fig. 1, table 1, and supplementary table 2, Supplementary Material online). The impact of hardware choices illustrates a trade-off between running time and carbon footprints, and is discussed in more detail below (see Parallelization and Processors). It should be noted that the running time of BEAST, and therefore its carbon footprint, scales as a power law, that is, not linearly, with the number of loci (Ogilvie et al. 2016).

We also estimated the carbon footprint of two large-scale empirical phylogenetic studies that each used over 300,000 CPU hours (table 1) (Jarvis et al. 2014; Zhou et al. 2018). As both studies were lacking hardware information, we assumed a CPU power draw of 12 W per core (the average from our database). Four different maximum likelihood-based phylogenetic programs were evaluated—RAxML (Stamatakis 2014) with ExaML (Kozlov et al. 2015), PhyML (Guindon and Gascuel 2003; Guindon et al. 2010), IQ-TREE (Nguyen et al. 2015), and FastTree (Price et al. 2010)—by conducting more than 670,000 tree inferences on 19 empirical phylogenomic data sets with thousands of genes and around 200 taxa. We estimated this would have a carbon footprint of 3,565 $kgCO_2e$ (3,889 tree-months or 324 tree-years). Additionally, using the maximum likelihood program ExaML, Jarvis et al. (2014) performed a 322-million-bp MULTIZ alignment of putatively orthologous genome regions across 48 species of *Neoaves* and had a similarly large carbon footprint of 4,372 $kgCO_2e$ (4,769 tree-months).

RNA Sequencing

RNA sequencing is the sequencing and analysis of all RNA in a sample. We first assessed the read alignment step in RNAseq using an extensive benchmarking where Baruzzo et al. (2017) looked at different data sets of 10 million 100-base paired-end strand-specific simulated reads of two different genomes, *Homo sapiens* (hg19) and *Plasmodium falciparum* (Baruzzo et al. 2017), which have substantially differing levels of complexity (*P. falciparum* has higher rates of polymorphisms and errors). We estimated the carbon footprint of aligning two sets of reads, one to each genome (T1 human and T3 Malaria). The three most-cited software tested, STAR (Dobin et al. 2013), HISAT2 (Kim et al. 2019, 2), and TopHat2 (Kim et al. 2013), all had low recall when aligning the malaria reads to the *P. falciparum* genome, so we also assessed Novoalign (NovoAlign | Novocraft n.d.) as it performed significantly better for this task (table 1). The carbon footprints ranged from 0.0052 to 0.67 $kgCO_2e$ for *P. falciparum*, with Novoalign having both the best performances and

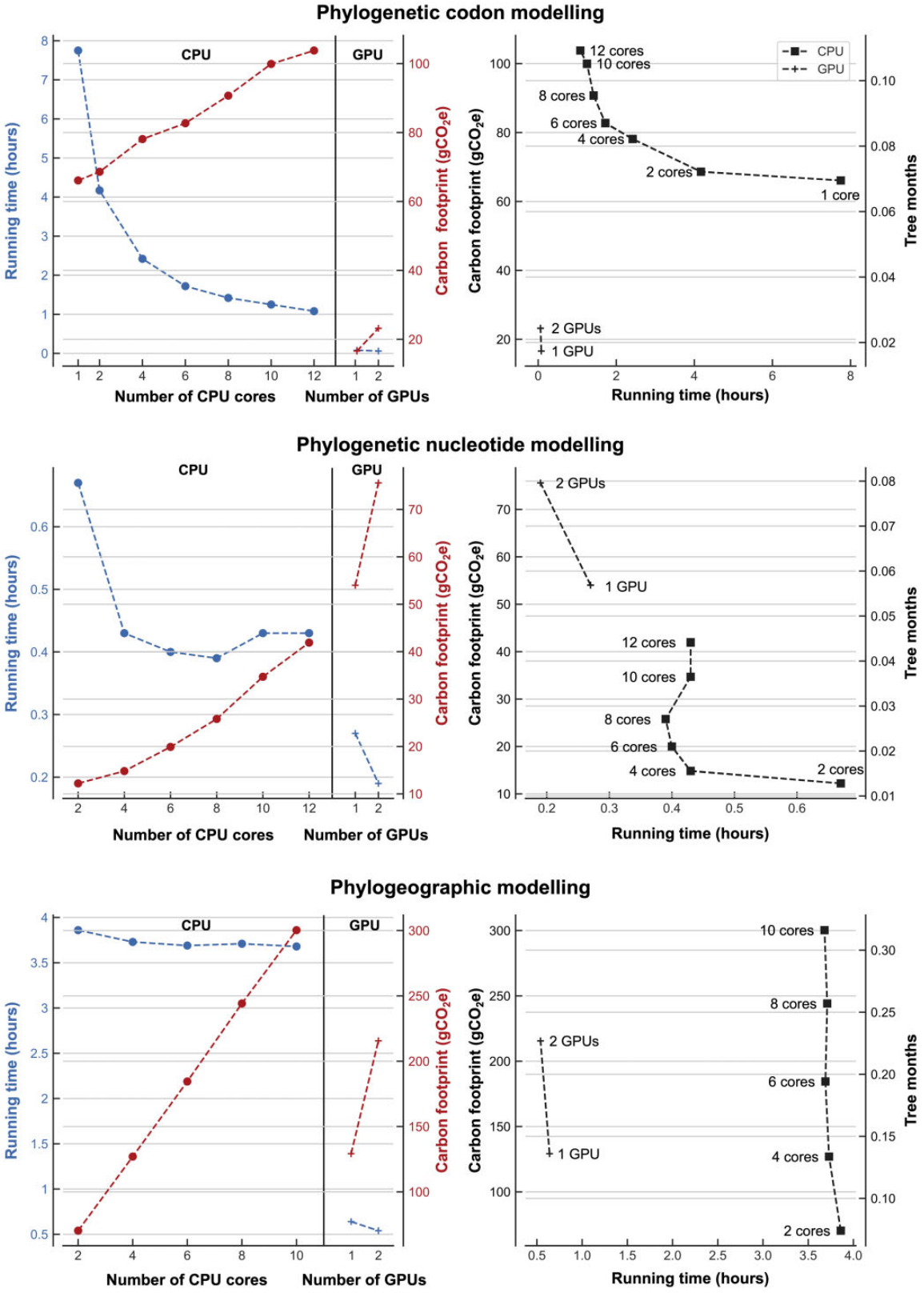

FIG. 1. The effect of hardware choices and parallelization on carbon footprint. The carbon footprint of BEAST/Beagle implemented on multicore CPU or GPUs for three different tasks. The plots on the left detail both the running time and carbon footprint against the number of cores utilized. The plots on the right detail the running time solely against carbon footprint (contextualized with tree-months) for both CPUs and GPUs. The numerical data are available in supplementary table 2, Supplementary Material online.

the largest carbon footprint. For human read alignment, despite all four methods obtaining high recall, their footprints varied by over two orders of magnitude (0.0054 to 0.98 $kgCO_2e$). As alignment tools are often reported with alignment speed (number of reads aligned in a given time) (Dobin et al. 2013; Kim et al. 2019, 2), the carbon footprints of the analyses above scale accordingly and ranged from 0.001 to 0.1 $kgCO_2e$ (0.001 to 0.1 tree-months) per million human or *P. falciparum* reads.

To quantify the carbon footprint of a full quality control pipeline with FastQC, we utilized 392 RNAseq read sets obtained from PBMC samples (Kusel et al. 2006, 2007), with a median depth of 45 million paired-end reads and average length 146 bp. Adapters were trimmed with TrimGalore(Babraham Bioinformatics – Trim Galore! n.d.), followed by the removal of optical duplicates using bbmap/clumpify (BBMap Guide n.d.). Reads were then aligned to the human genome reference (Ensemble GRCh 38.98) using STAR (Dobin et al. 2013). We estimated the carbon footprint of this pipeline to be 54.97 $kgCO_2e$ for the full data set, or 1.22 $kgCO_2e$ per million reads (table 1), which scales linearly with the number of reads (supplementary additional file 2, Supplementary Material online).

For transcript isoform abundance estimation, we assessed Sailfish (Patro et al. 2014), RSEM (Li and Dewey 2011), and Cufflinks (Trapnell et al. 2010) using the benchmark from Kanitz et al. (2015) on simulated human RNA-seq data (hg19). The Flux Simulator software (Griebel et al. 2012) and GENCODE (Harrow et al. 2012) were used to generate 100 million single-end 50-bp reads. The carbon footprints of this task were between 0.0081 and 1.40 $kgCO_2e$ (table 1), and the authors showed that the time complexity, and therefore the carbon footprint, is proportional to the number of reads. Additionally, these tools offer the option of parallelization, which can reduce running time but in this case, not carbon footprint; indeed, the decrease in running time when using 16 cores instead of one was not sufficient to offset the increase in power consumption, which resulted in a 2- to 6-fold increase in carbon footprint when utilizing 16 cores (table 1). There were significant differences between tools despite RSEM and Sailfish having similar accuracy performances in this benchmark. Since Sailfish does not perform a read alignment step and was on an average 53 times faster than RSEM, its carbon footprint was 71 times less than RSEM's when using 1 core and 39 times less with 16 cores. Lastly, although Cufflinks is largely used for abundance estimation, its main purpose is transcript isoform assembly, resulting in a significantly lower accuracy here (at a higher carbon cost).

Genome-Wide Association Analysis

Genome-wide association analysis aims to identify genetic variants across the genome associated with a phenotype. Here, we assessed both genome-wide association studies (GWAS) and expression qualitative trait locus (eQTL) mapping. We estimated the carbon footprint of GWAS with two different versions of Bolt-LMM (Loh et al. 2018) on the UK BioBank (Bycroft et al. 2018) (500k individuals, 93 M imputed SNPs). We found that a single trait GWAS would emit 17.29 $kgCO_2e$ with Bolt-LMM v1 and 4.70 $kgCO_2e$ with Bolt-LMM v2.3 (table 1), a reduction of 73%. GWAS typically assess multiple phenotypes, for example, metabolomics GWAS consider from several hundreds to several thousands of metabolites; since the association models in GWAS are typically fit on a per-trait basis, the carbon footprint is proportional to the number of traits analyzed. Bolt-LMM's carbon footprint also scales linearly with the number of genetic variants (BOLT-LMM v2.3.4 User Manual 2019), meaning that a single biobank-scale GWAS using UK Biobank (500k individuals) has a carbon footprint of 0.05 $kgCO_2e$ per million variants (0.06 tree-months) with Bolt-LMM v2.3 and 0.2 $kgCO_2e$ per million variants (0.2 tree-months) with Bolt-LMM v1. However, Bolt-LMM does not scale linearly with the number of samples ($time \sim O(N^{1.5})$; BOLT-LMM v2.3.4 User Manual 2019), which must be taken into account when scaling the values to a different sample size.

For cis-eQTL mapping, we compared the carbon footprint using either CPUs or GPUs on two data sets, first on a small sample size using skeletal muscle data from GTEx (GTEx Consortium 2017) (1 gene, 700 individuals) with a benchmark of FastQTL (CPU) (Ongen et al. 2016) and TensorQTL (GPU) (Taylor-Weiner et al. 2019; Broadinstitute/Tensorqtl (2018) 2020) from Taylor-Weiner et al. (2019). Besides, both tools were shown to yield similar mappings. Secondly, we used an in-house assessment (see Materials and Methods), to estimate the carbon footprint of a CPU-based analysis with LIMIX (Lippert et al. 2014) and with the GPU-based TensorQTL, using a larger cohort of 2,745 individuals with 18k genetic features and 10.7 m SNPs (table 1). In both cases, footprints were lower (28x and 94x) when using GPUs instead of CPUs. The scaling of eQTLs is complex, and the carbon footprint does not scale linearly with the number of traits or sample size (Lippert et al. 2014; Taylor-Weiner et al. 2019).

Molecular Simulations and Virtual Screening

Molecular simulations and virtual screening use computational simulations to model and understand molecular behavior and in silico scanning of small molecules for drug discovery. We estimated the carbon footprint of simulating molecular dynamics of the Satellite Tobacco Mosaic Virus (1,066,628 atoms) for 100 ns (nanoseconds) using AMBER and NAMD (NAMD Performance n.d.; The Pmemd.Cuda GPU Implementation n.d.) (Case et al. 2005; Phillips et al. 2005) and obtained between 18 and 95 $kgCO_2e$, which corresponds to 0.2 to 1 $kgCO_2e$ per ns (table 1). It should be noted that there are small discrepancies between the simulation parameters used by the tools so they cannot be compared directly (table 1), and due to a lack of information, neither of these estimations include the power usage from memory.

Using a benchmark from Ruiz-Carmona et al. (2014), we estimated the carbon footprint of three molecular docking methods, AutoDock Vina, Glide, and rDock (Friesner et al. 2004; Trott and Olson 2010; Ruiz-Carmona et al. 2014). The data originate from four systems (ADA, COMT, PARP, and

Trypsin) from the Directory of Useful Decoys benchmark set (Huang et al. 2006). To estimate their carbon footprints, we used the average computational running times for a 1 million ligand campaign and found values ranging from 13 to 514 $kgCO_2e$ (table 1). Glide was the fastest tool and had the smallest footprint, although it is not freely available. Of the two freely available tools (AutoDock Vina and rDock), rDock had the smallest carbon footprint with a performance comparable to Glide (Ruiz-Carmona et al. 2014).

Local versus Cloud Data Center, and the Role of Geography

Cloud computing facilities and large data centers are optimized to significantly reduce overhead power consumption such as cooling and lighting, and as such are often more energy efficient than smaller facilities. A report from 2016 estimated for example that energy usage by data centers in the United States could be reduced by 25% if 80% of the smaller data centers were aggregated into larger and more efficient data centers (hyperscale facilities) (Shehabi et al. 2016). Compared with the global average PUE of 1.67, Google Cloud's average PUE of 1.11 (Efficiency – Data Centers – Google n.d.) reduces the carbon footprint of a task by 34%. Other cloud providers also achieve low PUEs, Microsoft Azure reduces the carbon footprint by 33% (PUE = 1.125; Microsoft 2015) and Amazon Web Service by 28% (PUE = 1.2; AWS & Sustainability n.d.).

The use of cloud facilities may also enable further reductions of carbon footprint by allowing users to choose a geographic location with relatively low CI. As an example, we found that a typical GWAS of UK Biobank considering 100 traits using the aforementioned GWAS framework (see Genome-Wide Association Analysis) together with BoltLMM v2.3 on a Google Cloud server in the UK would lower the carbon footprint by 81% when compared with the average local data center in Australia (fig. 2), potentially saving 705 $kgCO_2e$ (769 tree-months, or 64 tree-years). To find the optimal strategy for specific analysis and facilities, it is best to directly use the Green Algorithm calculator (www.green-algorithms.org, last accessed 2022).

Parallelization

It is common practice to use parallelization to share the workload between several computing cores and reduce the total running time. However, it has been shown that this can increase carbon footprint (Lannelongue et al. 2021) and we found that parallelization frequently results in trade-offs between running time and carbon footprint. A general optimal solution to this trade-off is difficult to find as the relationship between carbon footprint and number of cores used may not be linear depending on the power management strategy of the servers. For modeling purposes, we assume here that cores are allocated independently to different users and that each core is used at 100%.

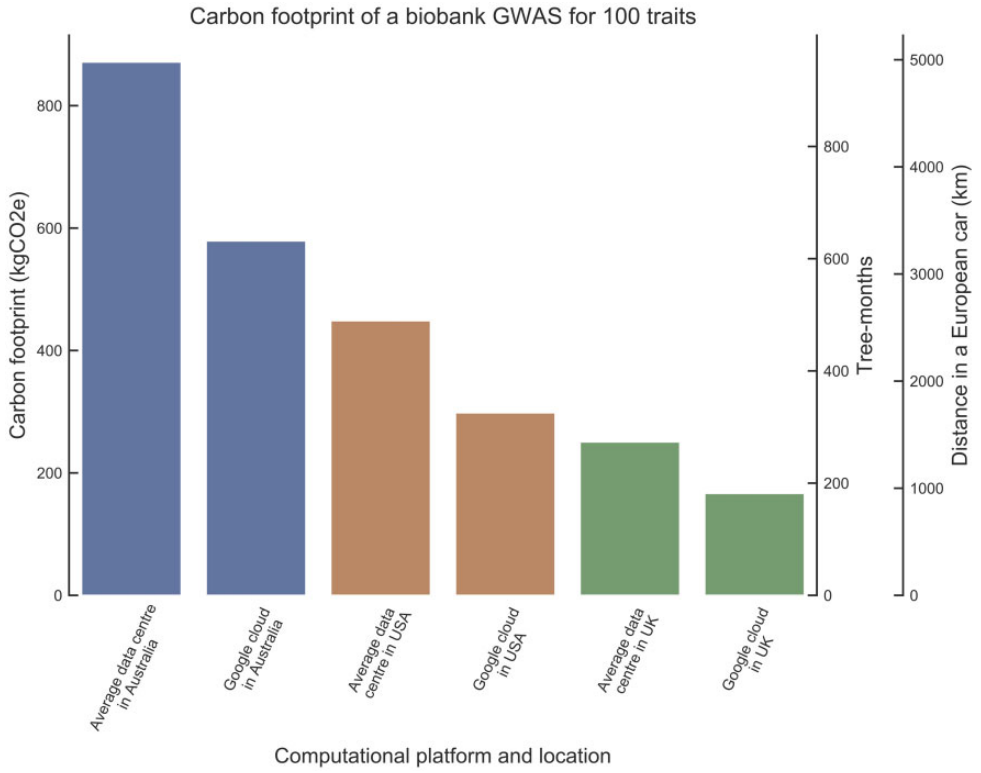

FIG. 2. Impact of location and computational platform on carbon footprint. Carbon footprint (in $kgCO_2e$, tree-months, and European car km) of a biobank scale 100 trait GWAS in various locations and platforms. Average data centers have a PUE of 1.67 (Andy 2019), Google cloud has a PUE of 1.11 (Efficiency – Data Centers – Google n.d.), Australia has a CI of 0.88 $kgCO_2e$/kWh, the United States 0.453 $kgCO_2e$/kWh, and the UK 0.253 $kgCO_2e$/kWh (Carbonfootprint.Com – International Electricity Factors 2020).

In some cases, the reduction in running time is substantial. For example, executing the phylogenetic codon model (see Phylogenetics) on a single core would take 7.8 h and emit 0.066 $kgCO_2e$, but with two cores, the carbon footprint increased by only 4% while running time was decreased by 46% (1.9x speedup) (fig. 1 and supplementary table 2, Supplementary Material online). With 12 cores, running time decreased 86% (7.2x speedup) but the carbon footprint increased by 57%. In other cases, speedup was marginal, making the added GHG emissions unnecessary. For example, the phylogeographic model had a running time of 3.86 h with a carbon footprint of 0.070 $kgCO_2e$ when using two cores; increasing to ten cores reduced running time by only 5% but increased carbon footprint by 4-fold (fig. 1 and supplementary table 2, Supplementary Material online).

The Impact of Memory

Provided memory is mobilized and not idle, its power consumption depends mainly on the memory available, not on the memory used (Karyakin and Salem 2017; Lannelongue et al. 2021). Thus, having too much memory available for a task results in unnecessary energy usage and GHG emissions. Although memory is usually a fixed parameter when working with a desktop computer or a laptop, on most computational servers and cloud platforms, the user can choose the memory allocated. Given it is common practice to over-allocate memory out of caution, we modeled the impact of memory allocation on carbon footprint in bioinformatics (fig. 3 and supplementary table 1, Supplementary Material online).

We showed that, while increasing the allocated memory always increases the carbon footprint, the effect is particularly significant for tasks with large memory requirements (fig. 3 and supplementary table 1, Supplementary Material online). For example, in de novo human genome assembly, MEGAHIT had higher memory requirements than ABySS (6% vs. 1% of total energy consumption); as a result, a 5-fold over-allocation of memory increases carbon footprint by 30% for MEGAHIT and 6% for ABySS. Similarly, in human RNA read alignment (fig. 3 and supplementary table 1, Supplementary Material online), Novoalign had the highest memory requirements (37% of its total energy vs. less than 7% for STAR, HISAT2, and TopHat2) and a 5x over-allocation in memory would increase its footprint by 187% compared with 32% for STAR, 2% for HISAT2, and 10% for TopHat2.

Processors

We estimated the carbon footprint of algorithms executed on both GPUs and CPUs. For cis-eQTL mapping (see Genome-Wide Association Analysis), we estimated that, compared with CPU-based FastQTL and LIMIX, using a GPU-based software like TensorQTL can reduce the carbon footprint by 96% and 99% and the running time by 99.63% and 99.99%, respectively (table 1). For the codon modeling benchmark (see Phylogenetics), utilizing GPUs had a speedup factor of 93x and 13x when compared with 1 and 12 CPU cores, resulting in a decrease in carbon footprint of 75% and 84% respectively. These estimations demonstrate that GPUs can be well suited to both reducing running time and carbon footprint for algorithms.

However, there are situations where the use of GPUs can increase carbon footprint. Using a GPU for phylogenetic nucleotide modeling (see Phylogenetics), instead of 8 CPU cores, decreased running time by 31% but also doubled the carbon footprint. We estimated that a single GPU would need to run the model in under 4 min to match the CPU's carbon footprint, as opposed to the 16 min it currently takes. Similarly, using a GPU for the phylogeographic modeling of the Ebola virus data set (see Phylogenetics) reduced the running time by 83% (6x speedup) when compared with the method with the lowest footprint (2 CPU cores) but increased carbon footprint by 84%. The equations used for this estimation are in supplementary note 1, Supplementary Material online, but a simple approximation can be used by scaling the running time of the GPU by the ratio of the power draws of the CPU and GPU. For example, we compared the popular Xeon E5-2683 CPU (using all 16 cores, power draw of 120 W) to the Tesla V100 GPU (300 W) and found that, to have the same carbon footprint with both configurations, an algorithm needs to run approximately 2.5 times (300/120) faster on GPU than CPU.

Discussion

In this work, we estimated the carbon footprint of various bioinformatic algorithms. Additionally, we investigated how memory over-allocation, processor choice, and parallelization affect carbon footprints, and showed the impact of transferring computations to cloud facilities.

This study made a series of important findings:

(1) For the same task, there can be orders of magnitude differences between the carbon footprints of the tools available, despite similar performances. This highlights the importance of factoring in GHG emissions when choosing a software.

(2) Limiting parallelization can reduce carbon footprints. Especially when the running time reduction is marginal, the carbon cost of parallelization should be closely examined. Besides, such methods to obtain faster running times may encourage scientists to run more computations; this rebound effect can increase carbon footprints further.

(3) Despite being often faster, GPUs do not necessarily have a smaller carbon footprint than CPUs, and it is useful to assess whether the running time reduction is large enough to offset the additional power consumption. In particular, when new hardware needs to be acquired, the environmental impact of manufacturing it should be taken into account.

(4) Using energy-efficient data centers, either local or cloud-based, can reduce carbon footprints by approximately 34% on an average.

(5) Substantial reductions in carbon footprint can be made by performing computations in energy-efficient countries with low CI.

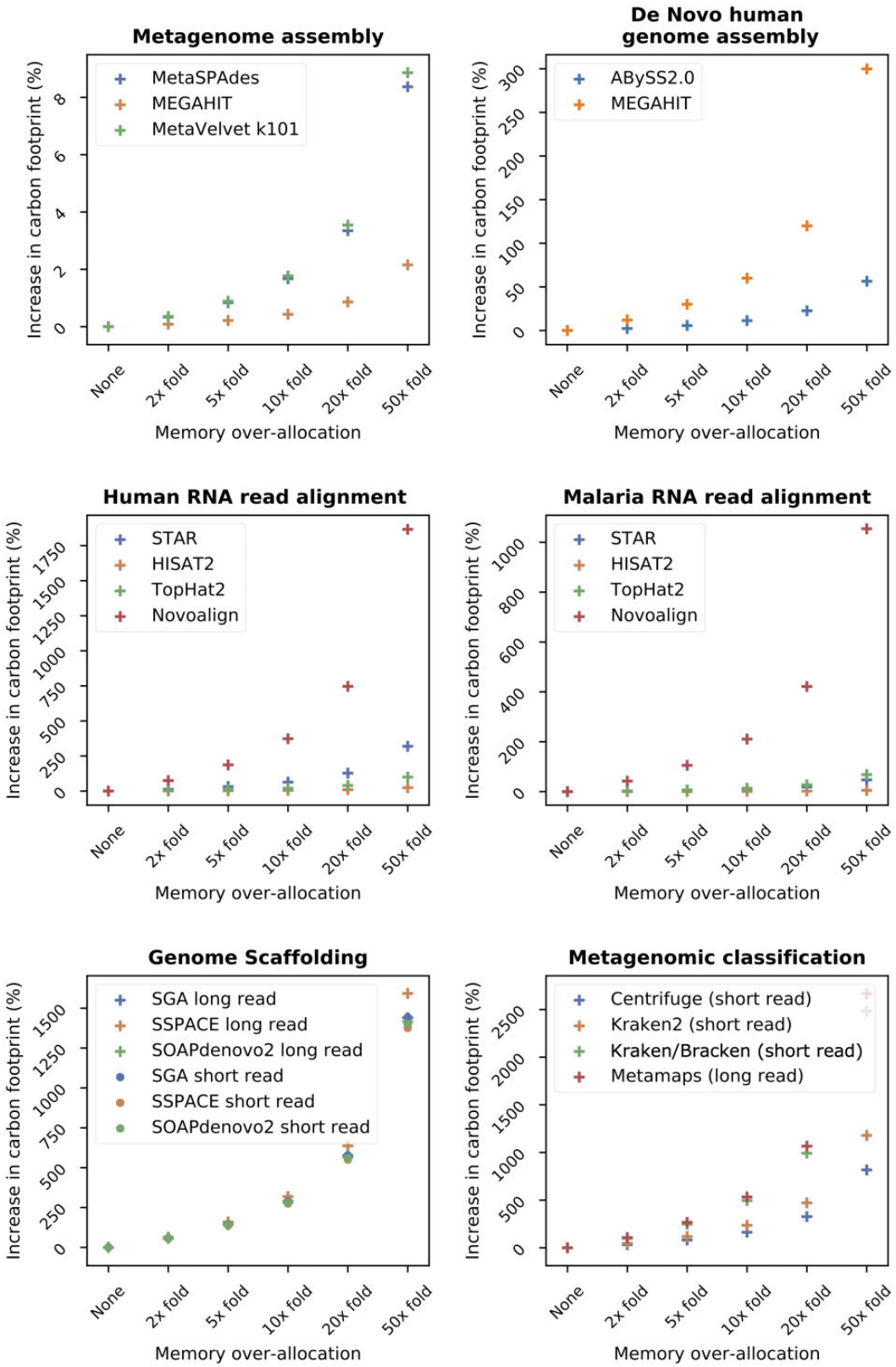

FIG. 3. Over-allocating memory increases a given algorithm's carbon footprint. We modeled how over-allocating the memory for a given algorithm increases its carbon footprint and this effect is increased for algorithms with larger memory requirements. Each plot details the percentage increase in carbon footprint as a function of memory overestimation for a variety of bioinformatic tools and tasks. The numerical data are available in supplementary table 1, Supplementary Material online.

(6) Carbon offsetting, which consists of supporting GHG-reducing projects, can be a way to balance the GHG emissions of computations. Although a number of cloud providers take part in this (AWS & Sustainability n.d.; Google Cloud Environment | Go Green n.d.; Global Infrastructure | Microsoft Azure n.d.), the real impact of carbon offsetting is debated and reducing the amount of GHG emitted in the first place should be prioritized.

(7) Over-allocating memory resources can unnecessarily, and significantly, increase the carbon footprint of a task, particularly if this task has high memory usage already. To decrease energy waste, one should allocate memory in a mindful manner and mobilize the minimum amount of memory needed for the task, while being careful not to under allocate memory either, as failed jobs are another source of energy waste. The modeling of the impact of overallocation here is based on a number of assumptions regarding memory power draw (Desrochers et al. 2016; Karyakin and Salem 2017) and orders of magnitude rather than exact values should be remembered. Additionally, software could be optimized to minimize memory requirements, potentially moving some aspects to disk where energy usage is far lower. However, this introduces a trade-off between memory usage and running time, and developers need to identify the most sustainable option on a case-by-case basis.

(8) A simple way to reduce the carbon footprint of a given algorithm is to use the most up to date software. We showed that updating a common GWAS software reduced carbon footprint by 73%, indicating that this may be the quickest, easiest, and potentially most impactful way to reduce one's carbon footprint.

There are a number of assumptions made when estimating the energy usage and carbon footprint of a given algorithm. These assumptions, and the associated limitations, have been discussed in detail within Lannelongue et al. (2021). In particular, we had to assume that processors were fully used (usage factor of 1) during the task, which is likely to slightly overestimate energy usage. Another noteworthy limitation of the work here is that many of the carbon footprints estimated are for a single run of any given tool; however, most algorithms have parameters that must be fine-tuned through trial and error, frequently extensively so. For example, in GWAS, various adjustments are made to the initial association analysis to reduce nonbiological variation, such as different phenotype normalizations, batch-effect correction, and ancestry-effect adjustments. Each of these adjustments multiplies the analysis' total carbon footprint and therefore the real GHG emissions are likely to be orders of magnitude greater than reported here.

There are other areas of computational biology, such as imaging or artificial intelligence analyses, that are not estimated here but are likely have substantial carbon footprints. Similarly, there are a number of other popular bioinformatics algorithms that have not been estimated within this study, examples include BLAST (Altschul et al. 1990), GROMACS (Spoel et al. 2005), and GATK (McKenna et al. 2010). Finally, it is generally the case that at least some parameters needed to estimate the carbon footprint are missing from published articles, for example, running time, hardware information, or software versions. If we are to fully understand the carbon footprint of the field of bioinformatics, or any computational research, it is crucial that this information is reported systematically (processor running time, memory usage, hardware, and software information) and that authors estimate their own carbon footprint using reliable tools.

This study is, to the best of our knowledge, the first to estimate the carbon footprint of common bioinformatics tools. We also investigated how parallelization, memory over-allocation, and hardware choices affect GHG emissions and showed that they could be reduced by utilizing efficient computing facilities. Finally, we outlined a range of ways bioinformaticians can use to may their carbon footprint.

Materials and Methods

Selection of Bioinformatic Tools

We estimated the carbon footprint of a range of tasks across the field of bioinformatics: genome and metagenome assembly, long and short reads metagenomic classification, RNA-seq and phylogenetic analyses, GWAS, eQTL mapping algorithms, molecular simulations, and molecular docking (table 1). For each task, we curated the published literature to identify peer-reviewed studies which computationally benchmarked popular tools. To be selected, publications had to report at least the running time and preferably memory usage and hardware used for the experiments, in particular the model and number of processing cores. We selected ten publications for this study (table 1). Besides, as we could not find suitable benchmarks to estimate the carbon footprint of cohort-scale eQTL mapping and RNA-seq quality control pipelines, we estimated the carbon footprint of these tasks using in-house computations. These computations were run on the Baker Heart and Diabetes Institute's computing cluster (Intel Xeon E5-2683 v4 CPUs and a Tesla T4 GPU) and the University of Cambridge's CSD3 computing cluster (Tesla P100 PCIe GPUs and Xeon Gold 6142 CPUs). In addition to estimating the carbon footprint, where possible, we provided estimations on how these footprints scale as the inputs vary.

Estimating the Carbon Footprint

The carbon footprint of a given tool was calculated using the framework described in Lannelongue et al. (2021) and the corresponding online calculator www.green-algorithms.org (last accessed 2022). We present here an overview of the methodology.

Electricity production emits a variety of GHGs, each with a different impact on climate change. To summarize this, the carbon footprint is measured in kilograms of CO_2-equivalent (CO_2e), which is the amount of carbon dioxide with an equivalent global warming impact as a mix of GHGs. This indicator

depends on two factors: the energy needed to run the algorithm, and the global warming impact of producing such energy, called CI. This can be summarized by:

$$C = E \times \mathrm{CI}, \quad (1)$$

where C is the carbon footprint (in kilograms of CO_2e—$kgCO_2e$), E is the energy needed (in W), and CI is the carbon intensity (in $kgCO_2e/W$).

The energy needs of an algorithm are measured based on running time, processing cores used, memory deployed, and efficiency of the data center:

$$E = t \times (n_c \times P_c \times u_c + n_m \times P_m) \times \mathrm{PUE} \times 0.001, \quad (2)$$

where t is the running time (h), n_c is the number of computing cores, used at u_c%, the core usage factor (between 0 and 1), and each core drawing a power P_c (W). n_m is the size of memory available (GB), drawing a power P_m (W/GB). PUE is the power usage effectiveness of the data center.

The power drawn by a processor (CPU or GPU) is estimated by its thermal design power per core, which is provided by the manufacturer, and then scaled by the core usage factor u_c. The power draw from memory was estimated to be 0.3725 W/GB. The PUE represents how much extra energy is needed to run the computing facilities, mainly for cooling and lighting.

The CI varies between countries because of the heterogeneity in energy production methods, from 0.012 $kgCO_2e$/kWh in Switzerland to 0.88 $kgCO_2e$/kWh in Australia for example (Carbonfootprint.Com – International Electricity Factors 2020). In order to be location-agnostic in this study, we used the global average value (0.475 $kgCO_2e$/kWh; Emissions – Global Energy & CO2 Status Report 2019 – Analysis 2019), unless otherwise specified.

Supplementary Material

Supplementary data are available at *Molecular Biology and Evolution* online.

Acknowledgments

We thank Kim van Daalen for the fruitful discussions about the impact of climate change on human health. We also thank Dr Michelle Wille for their helpful insights. J.G. was supported by a La Trobe University Postgraduate Research Scholarship jointly funded by the Baker Heart and Diabetes Institute and a La Trobe University Full-Fee Research Scholarship. L.L. was supported by the University of Cambridge MRC DTP (MR/S502443/1). This work was supported by core funding from the: UK Medical Research Council (MR/L003120/1), British Heart Foundation (RG/13/13/30194; RG/18/13/33946), and NIHR Cambridge Biomedical Research Centre (BRC-1215-20014) (The views expressed are those of the authors and not necessarily those of the NIHR or the Department of Health and Social Care). J.M. is currently an employee of Genomics PLC. This work was also supported by Health Data Research UK, which is funded by the UK Medical Research Council, Engineering and Physical Sciences Research Council, Economic and Social Research Council, Department of Health and Social Care (England), Chief Scientist Office of the Scottish Government Health and Social Care Directorates, Health and Social Care Research and Development Division (Welsh Government), Public Health Agency (Northern Ireland), and British Heart Foundation and Wellcome. M.I. was supported by the Munz Chair of Cardiovascular Prediction and Prevention. This study was supported by the Victorian Government's Operational Infrastructure Support (OIS) program.

Data Availability

The data sets used to support the conclusions of this article are available in Supplementary Material online (supplementary additional file 1, Supplementary Material online). The calculator used to estimate the carbon footprint is available at https://green-algorithms.org/, the code is available at https://github.com/GreenAlgorithms/green-algorithms-tool, and the method behind it is described in Lannelongue et al. (2021).

References

Accelerating Detection of Disease – UK Research and Innovation. n.d. [cited 2020 Oct 27]. Available from: https://www.ukri.org/innovation/industrial-strategy-challenge-fund/accelerating-detection-of-disease/.

Air Pollution. 2016. World Health Organisation. [cited 2020 Oct 17]. Available from: https://www.who.int/westernpacific/health-topics/air-pollution.

Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. 1990. Basic local alignment search tool. *J Mol Biol*. 215(3):403–410.

Andrae A, Edler T. 2015. On global electricity usage of communication technology: trends to 2030. *Challenges* 6(1):117–157.

Andy L. 2019. Is pue actually going up? *Uptime Institute Blog* (blog). [cited May 15, 2019]. Available from: https://journal.uptimeinstitute.com/is-pue-actually-going-up/.

AWS & Sustainability. n.d. Amazon Web Services, Inc. [cited 2020 Jul 27]. Available from: https://aws.amazon.com/about-aws/sustainability/.

Ayres DL, Darling A, Zwickl DJ, Beerli P, Holder MT, Lewis PO, Huelsenbeck JP, Ronquist F, Swofford DL, Cummings MP, et al. 2012. BEAGLE: an application programming interface and high-performance computing library for statistical phylogenetics. *Syst Biol*. 61(1):170–173.

Babraham Bioinformatics – Trim Galore! n.d. [cited 2020 Jul 27]. Available from: https://www.bioinformatics.babraham.ac.uk/projects/trim_galore/.

Baele G, Ayres DL, Rambaut A, Suchard MA, Lemey P. 2019. High-performance computing in Bayesian phylogenetics and phylodynamics using BEAGLE. In: Anisimova Maria, editor. Evolutionary genomics: statistical and computational methods. p. 691–722. New York: Springer.

Baruzzo G, Hayer KE, Kim EJ, Di Camillo B, FitzGerald GA, Grant GR. 2017. Simulation-based comprehensive benchmarking of RNA-seq aligners. *Nat Methods*. 14(2):135–139.

BBMap Guide. n.d. DOE Joint Genome Institute. [cited 2020 Jul 27]. Available from: https://jgi.doe.gov/data-and-tools/bbtools/bb-tools-user-guide/bbmap-guide/.

Bender EM, Gebru T, McMillan-Major A, Shmitchell S. 2021. On the dangers of stochastic parrots: can language models be too big? Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency. Virtual Event Canada: ACM. p. 610–623.

Boetzer M, Henkel CV, Jansen HJ, Butler D, Pirovano W. 2011. Scaffolding pre-assembled contigs using SSPACE. *Bioinformatics* 27(4):578–579.

BOLT-LMM v2.3.4 User Manual. 2019. [cited 2020 Jul 23]. Available from: https://data.broadinstitute.org/alkesgroup/BOLT-LMM/#x1-150003.2.
Bradnam KR, Fass JN, Alexandrov A, Baranay P, Bechner M, Birol I, Boisvert S, Chapman JA, Chapuis G, Chikhi R, et al. 2013. Assemblathon 2: evaluating de novo methods of genome assembly in three vertebrate species. *Gigascience* 2(1):10.
Broadinstitute/Tensorqtl. 2018. 2020. Python. Cambridge (MA): Broad Institute. Available from: https://github.com/broadinstitute/tensorqtl.
Bycroft C, Freeman C, Petkova D, Band G, Elliott LT, Sharp K, Motyer A, Vukcevic D, Delaneau O, O'Connell J, et al. 2018. The UK Biobank resource with deep phenotyping and genomic data. *Nature* 562(7726):203–209.
Carbonfootprint.Com – International Electricity Factors. 2020. [cited 2021 Jan 21]. Available from: https://www.carbonfootprint.com/international_electricity_factors.html.
Case DA, Cheatham TE, Darden T, Gohlke H, Luo R, Merz KM, Onufriev A, Simmerling C, Wang B, Woods RJ. 2005. The AMBER biomolecular simulation programs. *J Comput Chem*. 26(16):1668–1688.
Desrochers S, Paradis C, Weaver VM. 2016. A validation of DRAM RAPL power measurements. Proceedings of the Second International Symposium on Memory Systems. MEMSYS '16. New York: Association for Computing Machinery. p. 455–470.
Dilthey AT, Jain C, Koren S, Phillippy AM. 2019. Strain-level metagenomic assignment and compositional estimation for long reads with MetaMaps. *Nat Commun*. 10(1):3066.
Dobin A, Davis CA, Schlesinger F, Drenkow J, Zaleski C, Jha S, Batut P, Chaisson M, Gingeras TR. 2013. STAR: ultrafast universal RNA-seq aligner. *Bioinformatics* 29(1):15–21.
Drummond AJ, Suchard MA, Xie D, Rambaut A. 2012. Bayesian phylogenetics with BEAUti and the BEAST 1.7. *Mol Biol Evol*. 29(8):1969–1973.
Dudas G, Carvalho LM, Bedford T, Tatem AJ, Baele G, Faria NR, Park DJ, Ladner JT, Arias A, Asogun D, et al. 2017. Virus genomes reveal factors that spread and sustained the Ebola epidemic. *Nature* 544(7650):309–315.
Efficiency – Data Centers – Google. n.d. Google Data Centers. [cited 2020 Jul 27]. Available from: https://www.google.com/about/datacenters/efficiency/.
Emissions – Global Energy & CO2 Status Report 2019 – Analysis. 2019. IEA. [cited 2020 Feb 10]. Available from: https://www.iea.org/reports/global-energy-co2-status-report-2019/emissions.
Friesner RA, Banks JL, Murphy RB, Halgren TA, Klicic JJ, Mainz DT, Repasky MP, Knoll EH, Shelley M, Perry JK, et al. 2004. Glide: a new approach for rapid, accurate docking and scoring. 1. Method and assessment of docking accuracy. *J Med Chem*. 47(7):1739–1749.
GTEx Consortium. Genetic Effects on Gene Expression across Human Tissues. 2017. *Nature* 550(7675):204–213.
Global Infrastructure | Microsoft Azure. n.d. [cited 2020 Jul 31]. Available from: https://azure.microsoft.com/en-us/global-infrastructure/.
Goldman N, Yang Z. 1994. A codon-based model of nucleotide substitution for protein-coding DNA sequences. *Mol Biol Evol*. 11(5):725–736.
Google Cloud Environment | Go Green. n.d. Google Cloud. [cited 2020 Jul 31]. Available from: https://cloud.google.com/sustainability.
Greenhouse Gas Reporting: Conversion Factors 2019. 2019. GOV.UK. [cited 2021 Feb 24]. Available from: https://www.gov.uk/government/publications/greenhouse-gas-reporting-conversion-factors-2019.
Griebel T, Zacher B, Ribeca P, Raineri E, Lacroix V, Guigó R, Sammeth M. 2012. Modelling and simulating generic RNA-seq experiments with the flux simulator. *Nucleic Acids Res*. 40(20):10073–10083.
Guindon S, Dufayard J-F, Lefort V, Anisimova M, Hordijk W, Gascuel O. 2010. New algorithms and methods to estimate maximum-likelihood phylogenies: assessing the performance of PhyML 3.0. *Syst Biol*. 59(3):307–321.
Guindon S, Gascuel O. 2003. A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. *Syst Biol*. 52(5):696–704.
Harrow J, Frankish A, Gonzalez JM, Tapanari E, Diekhans M, Kokocinski F, Aken BL, Barrell D, Zadissa A, Searle S, et al. 2012. GENCODE: the reference human genome annotation for the ENCODE project. *Genome Res*. 22(9):1760–1774.
Helmers E, Leitão J, Tietge U, Butler T. 2019. CO2-equivalent emissions from European passenger vehicles in the years 1995–2015 based on real-world use: assessing the climate benefit of the European 'Diesel Boom'. *Atmos Environ*. 198:122–132.
Huang N, Shoichet BK, Irwin JJ. 2006. Benchmarking sets for molecular docking. *J Med Chem*. 49(23):6789–6801.
Hunt M, Newbold C, Berriman M, Otto TD. 2014. A comprehensive evaluation of assembly scaffolding tools. *Genome Biol*. 15(3):R42.
ICGC/TCGA Pan-Cancer Analysis of Whole Genomes Consortium. 2020. Pan-cancer analysis of whole genomes. *Nature* 578(7793):82–93.
Jackman SD, Vandervalk BP, Mohamadi H, Chu J, Yeo S, Hammond SA, Jahesh G, Khan H, Coombe L, Warren RL, et al. 2017. ABySS 2.0: resource-efficient assembly of large genomes using a bloom filter. *Genome Res*. 27(5):768–777.
Jahnke K, Fendt C, Fouesneau M, Georgiev I, Herbst T, Kaasinen M, Kossakowski D, Rybizki J, Schlecker M, Seidel G, et al. 2020. An astronomical institute's perspective on meeting the challenges of the climate crisis. *Nat Astron*. 4(9):812–815.
Jarvis ED, Mirarab S, Aberer AJ, Li B, Houde P, Li C, Ho SYW, Faircloth BC, Nabholz B, Howard JT, et al. 2014. Whole-genome analyses resolve early branches in the tree of life of modern birds. *Science* 346(6215):1320–1331.
Jones N. 2018. How to stop data centres from gobbling up the world's electricity. *Nature* 561(7722):163–166.
Kachuri L, Graff RE, Smith-Byrne K, Meyers TJ, Rashkin SR, Ziv E, Witte JS, Johansson M. 2020. Pan-cancer analysis demonstrates that integrating polygenic risk scores with modifiable risk factors improves risk prediction. *Nat Commun*. 11(6084):1–11.
Kanitz A, Gypas F, Gruber AJ, Gruber AR, Martin G, Zavolan M. 2015. Comparative assessment of methods for the computational inference of transcript isoform abundance from RNA-seq data. *Genome Biol*. 16(1):150.
Karyakin A, Salem K. 2017. An analysis of memory power consumption in database systems. Proceedings of the 13th International Workshop on Data Management on New Hardware – DAMON '17. Chicago, Illinois: ACM Press. p. 1–9.
Kim D, Paggi JM, Park C, Bennett C, Salzberg SL. 2019. Graph-based genome alignment and genotyping with HISAT2 and HISAT-genotype. *Nat Biotechnol*. 37(8):907–915.
Kim D, Pertea G, Trapnell C, Pimentel H, Kelley R, Salzberg SL. 2013. TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome Biol*. 14(4):R36.
Kim D, Song L, Breitwieser FP, Salzberg SL. 2016. Centrifuge: rapid and sensitive classification of metagenomic sequences. *Genome Res*. 26(12):1721–1729.
Kozlov AM, Aberer AJ, Stamatakis A. 2015. ExaML version 3: a tool for phylogenomic analyses on supercomputers. *Bioinformatics* 31(15):2577–2579.
Kusel MMH, de Klerk NH, Holt PG, Kebadze T, Johnston SL, Sly PD. 2006. Role of respiratory viruses in acute upper and lower respiratory tract illness in the first year of life: a birth cohort study. *Pediatr Infect Dis J*. 25(8):680–686.
Kusel MMH, de Klerk NH, Kebadze T, Vohma V, Holt PG, Johnston SL, Sly PD. 2007. Early-life respiratory viral infections, atopic sensitization, and risk of subsequent development of persistent asthma. *J Allergy Clin Immunol*. 119(5):1105–1110.
Lannelongue L, Grealey J, Inouye M. 2021. Green algorithms: quantifying the carbon footprint of computation. *Adv Sci (Weinh)*. 8(12):2100707.
Li B, Dewey CN. 2011. RSEM: accurate transcript quantification from RNA-seq data with or without a reference genome. *BMC Bioinformatics* 12:323.
Li D, Luo R, Liu C-M, Leung C-M, Ting H-F, Sadakane K, Yamashita H, Lam T-W. 2016. MEGAHIT v1.0: a fast and scalable metagenome assembler driven by advanced methodologies and community practices. *Methods* 102:3–11.

Lippert C, Paolo Casale F, Rakitsch B, Stegle O. 2014. LIMIX: genetic analysis of multiple traits. bioRxiv. Available from: https://www.biorxiv.org/content/10.1101/003905v2

Lischer HEL, Shimizu KK. 2017. Reference-guided de novo assembly approach improves genome reconstruction for related species. *BMC Bioinformatics* 18(1):474.

Loh P-R, Kichaev G, Gazal S, Schoech AP, Price AL. 2018. Mixed-model association for Biobank-scale datasets. *Nat Genet*. 50(7):906–908.

Lu J, Breitwieser FP, Thielen P, Salzberg SL. 2017. Bracken: estimating species abundance in metagenomics data. *PeerJ Comput Sci*. 3:e104.

Luo R, Liu B, Xie Y, Li Z, Huang W, Yuan J, He G, Chen Y, Pan Q, Liu Y, et al. 2012. SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler. *GigaScience* 1(1):18.

McKenna A, Hanna M, Banks E, Sivachenko A, Cibulskis K, Kernytsky A, Garimella K, Altshuler D, Gabriel S, Daly M, et al. 2010. The genome analysis toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res*. 20(9):1297–1303.

Microsoft. 2015. Microsoft's cloud infrastructure, datacenters and network fact sheet. Redmond (WA): Microsoft Corporation, One Microsoft Way. Available from: http://download.microsoft.com/download/8/2/9/8297f7c7-ae81-4e99-b1db-d65a01f7a8ef/microsoft_cloud_infrastructure_datacenter_and_network_fact_sheet.pdf.

NAMD Performance. n.d. [cited 2020 Jul 25]. Available from: https://www.ks.uiuc.edu/Research/namd/benchmarks/.

Namiki T, Hachiya T, Tanaka H, Sakakibara Y. 2012. MetaVelvet: an extension of velvet assembler to de novo metagenome assembly from short sequence reads. *Nucleic Acids Res*. 40(20):e155.

Nathans J, Sterling P. 2016. How scientists can reduce their carbon footprint. *ELife* 5:e15928.

National Institutes of Health (NIH) – All of Us. n.d. [cited 2020 Oct 27]. Available from: https://allofus.nih.gov/.

Nguyen L-T, Schmidt HA, von Haeseler A, Minh BQ. 2015. IQ-TREE: a fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. *Mol Biol Evol*. 32(1):268–274.

NovoAlign | Novocraft. n.d. [cited 2020 Nov 14]. Available from: http://www.novocraft.com/products/novoalign/.

Nurk S, Meleshko D, Korobeynikov A, Pevzner P. 2017. MetaSPAdes: a new versatile de novo metagenomics assembler. *Genome Res*. 27(5):824–834.

Ogilvie HA, Heled J, Xie D, Drummond AJ. 2016. Computational performance and statistical accuracy of BEAST and comparisons with other methods. *Syst Biol*. 65(3):381–396.

Ongen H, Buil A, Brown AA, Dermitzakis ET, Delaneau O. 2016. Fast and efficient QTL mapper for thousands of molecular phenotypes. *Bioinformatics* 32(10):1479–1485.

Patro R, Mount SM, Kingsford C. 2014. Sailfish enables alignment-free isoform quantification from RNA-seq reads using lightweight algorithms. *Nat Biotechnol*. 32(5):462–464.

PCAWG Structural Variation Working Group, PCAWG Consortium, Li Y, Roberts ND, Wala JA, Shapira O, Schumacher SE, Kumar K, Khurana E, Waszak S, et al. 2020. Patterns of somatic structural variation in human cancer genomes. *Nature* 578(7793):112–121.

Phillips JC, Braun R, Wang W, Gumbart J, Tajkhorshid E, Villa E, Chipot C, Skeel RD, Kalé L, Schulten K. 2005. Scalable molecular dynamics with NAMD. *J Comput Chem*. 26(16):1781–1802.

Pmemd.Cuda GPU Implementation. n.d. [cited 2020 Jul 23]. Available from: https://ambermd.org/GPUPerformance.php.

Portegies Zwart S. 2020. The ecological impact of high-performance computing in astrophysics. *Nat Astron*. 4(9):819–822.

Price MN, Dehal PS, Arkin AP. 2010. FastTree 2 – approximately maximum-likelihood trees for large alignments. *PLoS One* 5(3):e9490.

Primary Energy Consumption by World Region. 2021. Our World in Data. [cited 2021 Jan 25]. Available from: https://ourworldindata.org/grapher/primary-energy-consumption-by-region.

Ruiz-Carmona S, Alvarez-Garcia D, Foloppe N, Beatriz Garmendia-Doval A, Juhos S, Schmidtke P, Barril X, Hubbard RE, Morley SD. 2014. RDock: a fast, versatile and open source program for docking ligands to proteins and nucleic acids. *PLoS Comput Biol*. 10(4):e1003571.

Salzberg SL, Phillippy AM, Zimin A, Puiu D, Magoc T, Koren S, Treangen TJ, Schatz MC, Delcher AL, Roberts M, et al. 2012. GAGE: a critical evaluation of genome assemblies and assembly algorithms. *Genome Res*. 22(3):557–567.

Severe Covid-19 GWAS Group. 2020. Genomewide association study of severe COVID-19 with respiratory failure. *New Engl J Med*. 383(16):1522–1534.

Shehabi A, Smith S, Sartor D, Brown R, Herrlin M, Koomey J, Masanet E, Horner N, Azevedo I, Lintner W. 2016. United States data center energy usage report. Berkeley (CA): Lawrence Berkeley National Laboratory. LBNL–1005775, 1372902.

Simpson JT, Durbin R. 2012. Efficient de novo assembly of large genomes using compressed data structures. *Genome Res*. 22(3):549–556.

Spoel DVD, Lindahl E, Hess B, Groenhof G, Mark AE, Berendsen HJC. 2005. GROMACS: fast, flexible, and free. *J Comput Chem*. 26(16):1701–1718.

Stamatakis A. 2014. RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics* 30(9):1312–1313.

Stevens ARH, Bellstedt S, Elahi PJ, Murphy MT. 2020. The imperative to reduce carbon emissions in astronomy. *Nat Astron*. 4(9):843–851.

Strubell E, Ganesh A, McCallum A. 2019. Energy and policy considerations for deep learning in NLP. *ArXiv:1906.02243 [Cs]*, June. Available from: http://arxiv.org/abs/1906.02243.

Sutton TDS, Clooney AG, Ryan FJ, Ross RP, Hill C. 2019. Choice of assembly software has a critical impact on virome characterisation. *Microbiome* 7(1).

Taylor-Weiner A, Aguet F, Haradhvala NJ, Gosai S, Anand S, Kim J, Ardlie K, Van Allen EM, Getz G. 2019. Scaling computational genomics to millions of individuals with GPUs. *Genome Biol*. 20(1):228.

Trapnell C, Williams BA, Pertea G, Mortazavi A, Kwan G, van Baren MJ, Salzberg SL, Wold BJ, Pachter L. 2010. Transcript assembly and quantification by RNA-seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nat Biotechnol*. 28(5):511–515.

Trott O, Olson AJ. 2010. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization and multithreading. *J Comput Chem*. 31(2):455–461.

Vollmers J, Wiegand S, Kaster A-K. 2017. Comparing and evaluating metagenome assembly tools from a microbiologist's perspective – not only size matters! *PLoS One* 12(1):e0169662.

Watts N, Amann M, Arnell N, Ayeb-Karlsson S, Belesova K, Boykoff M, Byass P, Cai W, Campbell-Lendrum D, Capstick S, et al. 2019. The 2019 report of the lancet countdown on health and climate change: ensuring that the health of a child born today is not defined by a changing climate. *Lancet* 394(10211):1836–1878.

Wood DE, Lu J, Langmead B. 2019. Improved metagenomic analysis with Kraken 2. *Genome Biol*. 20(1):257.

Wood DE, Salzberg SL. 2014. Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome Biol*. 15(3):R46.

Zerbino DR, Birney E. 2008. Velvet: algorithms for de novo short read assembly using de Bruijn graphs. *Genome Res*. 18(5):821–829.

Zhou X, Shen X-X, Hittinger CT, Rokas A. 2018. Evaluating fast maximum likelihood-based phylogenetic programs using empirical phylogenomic data sets. *Mol Biol Evol*. 35(2):486–503.