

Accelerating genomic workflows using NVIDIA Parabricks

2

3 Kyle A. O'Connell¹, Zelaikha B. Yosufzai¹, Ross A. Campbell¹, Collin J. Lobb¹, Haley T.
4 Engelken¹, Laura M. Gorrell¹, Thad B. Carlson², Josh J. Catana¹, Dina Mikdadi¹, Vivien R.
5 Bonazzi^{1*}, Juergen A. Klenk^{1*}

6

⁷ ¹Biomedical Data Science Team, Deloitte Consulting LLP, Arlington VA, 20057

⁸ ²Cloud Managed Services, Deloitte Consulting LLP, Detroit MI, 48226

9

10 *Corresponding authors: Vivien R. Bonazzi, vbonazzi@deloitte.com; Juergen A. Klenk,
11 jklenk@deloitte.com

12

13 ABSTRACT

14 Background

15 As genome sequencing becomes a more integral part of scientific research, government policy,
16 and personalized medicine, the primary challenge for researchers is shifting from generating raw
17 data to analyzing these vast datasets. Although much work has been done to reduce compute
18 times using various configurations of traditional CPU computing infrastructures, Graphics
19 Processing Units (GPUs) offer the opportunity to accelerate genomic workflows by several
20 orders of magnitude. Here we benchmark one GPU-accelerated software suite called NVIDIA
21 Parabricks on Amazon Web Services (AWS), Google Cloud Platform (GCP), and an NVIDIA
22 DGX cluster. We benchmarked six variant calling pipelines, including two germline callers

23 (HaplotypeCaller and DeepVariant) and four somatic callers (Mutect2, Muse, LoFreq,
24 SomaticSniper).

25 **Results**

26 For germline callers, we achieved up to 65x acceleration, bringing HaplotypeCaller runtime
27 down from 36 hours to 33 minutes on AWS, 35 minutes on GCP, and 24 minutes on the
28 NVIDIA DGX. Somatic callers exhibited more variation between the number of GPUs and
29 computing platforms. On cloud platforms, GPU-accelerated germline callers resulted in cost
30 savings compared with CPU runs, whereas somatic callers were often more expensive than CPU
31 runs because their GPU acceleration was not sufficient to overcome the increased GPU cost.

32 **Conclusions**

33 Germline variant callers scaled with the number of GPUs across platforms, whereas somatic
34 variant callers exhibited more variation in the number of GPUs with the fastest runtimes,
35 suggesting that these workflows are less GPU optimized and require benchmarking on the
36 platform of choice before being deployed at production scales. Our study demonstrates that
37 GPUs can be used to greatly accelerate genomic workflows, thus bringing closer to grasp urgent
38 societal advances in the areas of biosurveillance and personalized medicine.

39 **Keywords**

40 GPU acceleration, NVIDIA Parabricks, Cloud Computing, Amazon Web Services, Google
41 Cloud Platform

42

43

44 **BACKGROUND**

45 As the cost of genome sequencing continues to decrease, genomic datasets grow in both
46 size and generation speed (Langmead & Nellore, 2018). These processes will greatly enhance
47 aims such as whole genome biosurveillance and personalized medicine (Nwadiugwu &
48 Monteiro, 2022; Zhao et al., 2020). However, one challenge to attaining these goals is the
49 computational burden of analyzing large amounts of genomic sequence data (Liu et al., 2014).
50 Two trends (among others) are helping to ameliorate this burden. The first is the migration to the
51 cloud for data analysis and storage, and the second is the use of Graphics Processing Units
52 (GPUs) to accelerate data processing and analysis (Cole & Moore, 2018); (Franke & Crowgey,
53 2020). We address each of these trends in this article.

54 Cloud computing addresses many of the challenges associated with large whole genome
55 sequencing projects, which can suffer from siloed data, long download times, and slow
56 workflow runtimes (Tanjø et al., 2021). Several papers have reviewed the potential of cloud
57 platforms for sequence data storage, sharing, and analysis (Augustyn et al., 2021; Cole & Moore,
58 2018; Grossman, 2019; Grzesik et al., 2021; Koppad et al., 2021; Langmead & Nellore, 2018;
59 Leonard et al., 2019), thus here we focus on one cloud computing challenge, how to select the
60 right compute configuration to optimize both cost and performance (Krissaane et al., 2020; Ray
61 et al., 2021).

62 GPU acceleration in either a cloud or High Performance Computing (HPC) environment
63 makes rapid genomic analysis possible at a scale previously not possible. While these are still
64 early days for GPU-acceleration in the ‘omics fields, several studies have begun benchmarking
65 various algorithmic and hardware configurations to find the balance between cost and
66 performance. Franke & Crowgey, (2020) and Rosati, (2020) both benchmarked GATK
67 HaplotypeCaller using the original CPU algorithm and the GPU-accelerated version from

68 NVIDIA Clara™ Parabricks (<https://www.parabricks.com/>; hereafter Parabricks) on HPC
69 platforms and found notable acceleration (8x and 21x speedups respectively) when using GPUs.
70 They also inferred high concordance of SNP calls (~99.5%) between the CPU and GPU
71 algorithms suggesting no loss of accuracy with the GPU-configured algorithms, for both
72 germline and somatic variant callers (*Benchmarking NVIDIA Clara Parabricks Somatic Variant*
73 *Calling Pipeline on AWS*, 2022). Likewise, Zhang et al., (2021) introduced a new GPU-
74 accelerated pipeline called BaseNumber, which achieved runtimes slightly faster than previous
75 benchmarks using Parabricks.

76 While the aforementioned studies conducted benchmarking using on-premises computing
77 clusters, some studies have begun benchmarking GPU-accelerated algorithms in the cloud. The
78 Parabricks team at NVIDIA benchmarked GATK HaplotypeCaller using Parabricks on Amazon
79 Web Services (AWS) and achieved runtimes as low as 28 minutes for a 30x genome with eight
80 A100 NVIDIA GPUs (*Benchmarking the NVIDIA Clara Parabricks Germline Pipeline on AWS*,
81 2021), and speedups ranging from 4x to 42x for somatic callers (*Benchmarking NVIDIA Clara*
82 *Parabricks Somatic Variant Calling Pipeline on AWS*, 2022). Relatedly, Krissaane et al., (2020)
83 benchmarked GWAS workflows using Spark Clusters (not NVIDIA Parabricks) on both Google
84 Cloud Platform (GCP) and Amazon Web Services (AWS) and found identical performance
85 between cloud platforms. While these studies have shed light on the performance of GATK
86 HaplotypeCaller using Parabricks, fewer studies have compared CPU and GPU performance for
87 additional germline and somatic variant callers, or compared performance across AWS, GCP and
88 an NVIDIA DGX cluster.

89 Here, we benchmark two germline variant callers and four somatic variant callers
90 comparing traditional x86 CPU algorithms with GPU-accelerated algorithms implemented with

91 NVIDIA Parabricks on AWS and GCP, and benchmark GPU-accelerated algorithms on an
92 NVIDIA DGX cluster. In the case of GPU-accelerated algorithms, we compare 2, 4, and 8 GPU
93 configurations. For germline callers, we observed speedups of up to 65x (GATK
94 HaplotypeCaller) and found that performance scaled linearly with the number of GPUs. We also
95 found that because GPUs run so quickly, researchers can save money by using them for germline
96 variant callers. Alternatively, somatic variant callers achieved speedups up to 56.8x for the
97 Mutect2 algorithm, but surprisingly, did not scale linearly with the number of GPUs,
98 emphasizing the need for algorithmic benchmarking before embarking on large-scale projects.

99

100 **RESULTS**

101

102 **CPU baseline across cloud platforms**

103 CPU machine performance varied considerably between AWS/GCP for most analyses.
104 For germline analyses, GCP performed faster for DeepVariant (18.8 hrs) compared with AWS
105 (22 hrs), whereas AWS performed faster for HaplotypeCaller (36.2 hrs) compared with GCP
106 (38.8 hrs; Table 1, Fig. 1). Somatic runtimes favored AWS, with the exception of Mutect2,
107 where GCP ran in 8.1 hrs compared with 16.9 hrs on AWS (Table 1, Fig. 1).

108

109 **GPU performance across cloud platforms**

110 For germline callers, 8-GPU runtimes were below 45 minutes for HaplotypeCaller and
111 DeepVariant across both cloud platforms. On AWS, we observed faster runtimes for the A100
112 compared with the V100 GPU machines (p4 vs p3 machine families), but the differences with 8
113 GPUs, where the number of CPUs were equal, were small for most workflows. Further,
114 comparisons between the 2 and 4 A100 GPU machines on GCP/AWS was not precise because

115 we were unable to limit the number of CPUs available for all workflows, thus differences in
116 times between the two cloud platforms were biased towards AWS for some algorithms.
117 Although the two germline workflows scaled linearly with the number of GPUs (Fig. 2), somatic
118 callers ran faster with 4 vs. 8 GPUs for
119 Muse on AWS (but not GCP), Mutect2 on AWS and GCP, and for SomaticSniper on AWS and
120 GCP (Fig. 2; S1). Compared with the CPU baselines, GPU runs on AWS (with A100 GPU) led
121 to acceleration of HaplotypeCaller up to 65.1x, DeepVariant up to 30.7x, Mutect2 up to 56.8x,
122 SomaticSniper up to 7.7x, Muse up to 18.9x, and Lofreq up to 3.7x (Table 1). On GCP, GPUs
123 resulted in acceleration of HaplotypeCaller up to 65.8x, DeepVariant up to 26.5x, Mutect2 up to
124 29.3x, SomaticSniper up to 7.0x, Muse up to 21.8x, and LoFreq up to 4.5x.

125 Although GPU machines are much more expensive than CPU machines, the accelerated
126 runtimes result in cost savings for most algorithms (Fig. 4). Leveraging GPUs on AWS with the
127 A100 machine resulted in cost savings up to 63% for HaplotypeCaller with 8 GPUs, 33% for
128 DeepVariant with 4 GPUs, and up to 57.6% for Mutect2 with 4 GPUs. Using the A100 GPU
129 machine resulted in even greater savings of 63% for HaplotypeCaller with 4 GPUs, 21% for
130 DeepVariant with 8 GPUs, and 80% for Mutect2 with 4 GPUs (Table S1).

131 On GCP GPU runs resulted in cost savings of up to 80.1% for HaplotypeCaller with 2
132 GPUs, 44.4% for DeepVariant with 4 GPUs, 71.6% for Mutect2 with 4 GPUs, 26.2% for
133 SomaticSniper with 2 GPUs, and up to 70.1% for Muse with 2 GPUs. However, on both
134 platforms, algorithms that were not well optimized actually cost much more to run with GPUs
135 rather than CPUs because the difference in runtimes was not enough to offset the extra GPU cost
136 (Fig. 4; S4). For example, CPU runs of LoFreq cost less than \$10/sample to run on both

137 platforms, but as much as \$30 with GPUs (Fig. S2). Likewise, CPU runs of Somatic Sniper cost
138 less than \$15 per sample on both platforms, but as much as \$75 on AWS with 8 GPUs.

139 For well optimized algorithms, results varied between variant callers on which numbers
140 of GPUs were the fastest (ranging from 2–8); subsequently cost savings reflect a balance
141 between speed and cost of a particular machine type that is not consistent between algorithms or
142 cloud providers. For example, A100 GPU runs were expensive on AWS because the
143 p4d.24xlarge machine type on demand price is \$32.8/hr, whereas the A100 machine type ranges
144 from \$12.24/hr for a 4 GPU machine, to \$24.5/hr for an 8 GPU machine. On GCP, the a2-
145 highgpu machine types range from \$7.4/hr (2 GPUs) to \$29.4.00/hr (8 GPUs). Alternatively,
146 CPU runs were slightly cheaper on AWS with an on demand price of \$1.36/hr compared with
147 \$1.75 on GCP. Prices here are given for the northern Virginia region calculated (at the time of
148 writing) using the pricing calculators from the respective cloud service providers. As time goes
149 on, these machine types will likely become less expensive with greater adoption.

150

151 **GPU performance on the DGX**

152 Germline workflows ran considerably faster on the DGX than on the cloud platforms, with
153 HaplotypeCaller finishing in 24.4 min and DeepVariant finishing in 27.1 min with 8 GPUs (Fig.
154 2; S1). Somatic variant callers were not faster in most cases than the cloud platforms, and in one
155 case, ran slower than on the cloud (Somatic Sniper; Fig. 2; S1). Interestingly, the pattern we
156 observed in the cloud where the 4 GPU runtimes were the fastest for Muse and Somatic Sniper
157 did not manifest on the DGX, where the 8 GPU runs were the fastest for all algorithms, with the
158 exception of Mutect2 (Fig. 2; S1). For Mutect2, the 4 GPU run was still the fastest on the DGX,
159 but the 8 GPU run was faster on the DGX than on both AWS/GCP (Fig. S1).

160 We also tested the effect of CPU number on performance of GPU runs. On AWS and
161 GCP the GPU machine types are preconfigured with 12 CPUs/1 GPU, but on the DGX we were
162 able to modify the number of CPUs for each run. We found that adding CPUs does decrease
163 runtimes (increase performance), but that reduction of runtimes plateaued after 48 CPUs (Fig.
164 S5).

165

166 **DISCUSSION**

167 The acceleration provided by GPU-accelerated algorithms confers several advantages to
168 researchers. First, GPU-acceleration enables researchers to rapidly run multiple algorithms
169 (Crowgey et al., 2021). Different variant callers exhibit biases leading to slightly different variant
170 calls (Zhao et al., 2020). Combining calls across algorithms can lead to higher accuracy, albeit
171 with a slightly higher type one error. Future studies could compare false positive and negative
172 rates for different strategies of combining calls across algorithms such as majority rule vs.
173 consensus site calls. Another advantage of GPU-accelerated genomic workflows is that they
174 allow researchers to process more samples with a fixed budget. Academic research programs are
175 often constrained by limited funding; the use GPU-acceleration may allow researchers to reduce
176 compute costs (and labor overhead) and thus process more samples for the same amount of
177 money. Finally, GPU-accelerated algorithms enable near-real-time decision making. Pathogen
178 biosurveillance benefits from rapid data processing to identify novel pathogens and allow
179 policymakers to act before an outbreak spreads (Gardy & Loman, 2018).

180

181 **Cloud platform considerations**

182 ***CPU only runs***

183 As more research programs migrate to cloud platforms, researchers will need to make
184 decisions about which platform provides the most advantages for both performance and cost
185 considerations. CPU runs were faster on the AWS c6i.8xlarge machine than on the GCP n2-32
186 for four algorithms, while DeepVariant and Mutect2 ran faster on GCP (Fig. 1). Both of these
187 machine types use the newest Intel Xeon Scalable processors (Ice Lake), but seem to have
188 inherent differences that would be difficult to identify without benchmarking particular
189 algorithms as we have done here. Regardless of cloud platform however, past work within our
190 research group showed that reduced runtimes driven by using the latest CPU processors
191 outweighs the increased per second cost (TC unpublished).

192 Another consideration that researchers should be aware of in the near term is that AWS is
193 migrating to newer ARM-based machine types, rather than x86 architectures. We had trouble
194 installing existing software on the ARM-based machines, and thus used the c6i.8xlarge machine.
195 This could present challenges for researchers in the future on AWS as the platform migrates
196 more machine types to ARM-based architectures, necessitating the rewriting and/or compiling of
197 common software. On GCP, we chose the N2 machine family as a balance between performance
198 and cost. GCP does offer the compute-optimized C2 machine family, which may run faster than
199 the N2 machines, but we did not benchmark those machines here.

200

201 ***GPU considerations on the cloud***

202 For germline workflows, AWS and GCP performed very similarly for both speed and
203 cost when using 8 A100 GPUs, although the 2 and 4 GPUs runs exhibited more variation (Fig.
204 2,4). In an effort to quantify the balance between cost and performance on each cloud platform,
205 we calculated a cost ratio metric by dividing the cost of the workflow by the xSpeedup for a

206 GPU run when compared to the CPU run for that workflow. Thus, a lower cost ratio indicates a
207 better value for a given GPU configuration (Table 1; Fig. 5). For the germline variant callers, the
208 best cost ratio on both platforms used 8 GPUs, and the ratio for AWS and GCP was similar
209 enough that we feel it should not impact the choice between cloud providers. For somatic variant
210 workflows, the best cost ratio was usually 2–4 GPUs, as these workflows were not well
211 optimized to use 8 GPUs on the cloud. Further, because LoFreq and Somatic Sniper were not
212 very accelerated with Parabricks, their high cost ratio suggests that it is not worth the extra cost
213 to run these workflows using GPUs. It should be noted that we only benchmarked using on
214 demand instances, and bioinformaticians could save additional costs by leveraging spot
215 instances.

216 GPU-accelerated bioinformatic workflows are still relatively new to the cloud, and as
217 such, not all tools are readily available everywhere. For example, at the time of writing,
218 Parabricks did not offer a Marketplace solution for GCP, although their team was working on
219 releasing one (G. Burnett *pers. comm*). Likewise, the Marketplace solution on AWS offered a
220 user-friendly way to access the Parabricks software suite without purchasing an annual license,
221 but this machine image did not support the p4 machine family with the A100 GPUs.
222 Nonetheless, although we were able to install Parabricks on the A100 machine on AWS, this
223 machine type was not readily available (at the time of writing) in most regions, and it was
224 difficult to procure this machine type to conduct our benchmarking. Perhaps using spot instances
225 would have been a better solution for these difficult to procure machine types. Finally, we
226 observed some decreases in runtime between the A100 and V100 GPU machines on AWS (Fig.
227 3). However, differences were relatively minor when using 8 GPUs – less than a minute for
228 DeepVariant and eight minutes for HaplotypeCaller. As long as the A100 machine type is

229 difficult to obtain and is not available with the Marketplace machine image, we recommend
230 using the V100 GPU machine without much cost to performance (Table 1, S1; Fig. S3).

231

232 **On-premises computing clusters**

233 For a myriad of reasons, some bioinformatic analyses will not migrate to the cloud, thus
234 requiring on-premises infrastructure. Although not every institution will have a DGX cluster
235 with A100 GPUs available, we show here that Parabricks runs well in an on-premises
236 environment. For those looking to achieve the fastest possible runtimes in a production
237 environment, the DGX ran considerably faster than AWS or GCP for germline callers, reducing
238 runtimes for HaplotypeCaller by 8 min and DeepVariant by 15 min, differences that could be
239 significant at large enough scales. We attribute these differences to the network communication
240 between GPUs and CPUs on the machines, which is better optimized on the DGX compared with
241 cloud-based instances, where GPUs may not be located in as close of proximity

242

243 **CONCLUSIONS**

244 We found that germline variant callers were well optimized with Parabricks and that GPU-
245 accelerated workflows can result in substantial savings of both time and costs. Alternatively,
246 somatic callers were accelerated, but exhibited substantial variation between algorithms, number
247 of GPUs, and computing platform, suggesting that benchmarking algorithms with a reduced
248 dataset is important before scaling up to an entire study. Though early days for GPU-accelerated
249 bioinformatic pipelines, ever faster computing processors bring us closer to important societal
250 aims such as tracking pathogens in near real-time to monitor emerging pandemics or enabling
251 milestones in the field of personalized medicine.

252

253 **MATERIALS AND METHODS**

254 **Sampling and Algorithms**

255 We benchmarked six variant callers for CPU and GPU speed and cost. We conducted all
256 benchmarking on the individual ‘HG002’ from the Genome in a Bottle Consortium (Krusche et
257 al., 2019; Zook et al., 2016) hosted by the National Institute of Standards and Technology, and
258 made available as part of the Precision FDA Truth Challenge V2
259 (<https://precision.fda.gov/challenges/10>). We downsampled the fastq files to 30x coverage using
260 Samtools (Li et al., 2009). We used Grch38 as our reference genome downloaded from the GATK
261 Reference Bundle. Our germline variant calling pipeline evaluated two germline variant callers:
262 HaplotypeCaller (Poplin, Chang, et al., 2018; Van der Auwera & O’Connor, 2020) and DeepVariant
263 (Poplin, Ruano-Rubio, et al., 2018). GPU benchmarking used Parabricks. For germline callers we
264 used ‘Germline Pipeline’ for GATK HaplotypeCaller, and for DeepVariant we used
265 ‘DeepVariant Germline Pipeline’. Each of these pipelines take fastq files as inputs and output
266 unfiltered variant call format (VCF) files. CPU benchmarking was conducted by writing custom
267 workflows using Snakemake (Mölder et al., 2021), following best practices for each tool and
268 exactly matching the workflows used by Parabricks (Data Accessibility).

269 Our somatic variant calling pipeline evaluated four somatic variant callers: Mutect2 (Van
270 der Auwera & O’Connor, 2020), SomaticSniper (Larson et al., 2012), Muse (Fan et al., 2016),
271 and LoFreq (Wilm et al., 2012). We generated synthetic somatic tumor data using SomatoSim
272 (Hawari et al., 2021). We added 198 single nucleotide polymorphisms (SNPs) at random variant
273 allele frequencies ranging from 0.001 to 0.4 (randomly generated using custom python scripts).
274 Sites were selected from the ICGC Data Portal ovarian cancer patient DO32536

275 (https://dcc.icgc.org/donors/DO32536?mutations=%7B%22size%22:50,%22from%22:151%7D).
276 We used the BAM file from the HaplotypeCaller pipeline (i.e., MarkDuplicates,
277 BaseRecalibration, and ApplyBQSR were run prior to the mutation process) as the input for
278 SomatoSim. For somatic variant callers, we used the Parabricks variant caller scripts
279 ('mutectcaller', 'somaticsniper_workflow', 'muse', 'lofreq') which take BAM files as input and
280 output VCF files. Each Parabricks tool was compared to a compatible CPU command as listed in
281 the Parabricks 3.7 documentation. We used Snakemake scripts as described for germline callers.
282 For benchmarking of MuSE, we used version v2.0 and set the number of threads to 1 to replicate
283 MuSE v1.0 lack of parallel computing because of version conflicts with MuSE v1 in our
284 compute environment. We created a conda environment before running each workflow because
285 we found that using the '--with conda' flag in Snakemake dramatically increased run times.
286 Complete workflows are described in the Supporting Information and all scripts necessary to
287 repeat our analyses are available at (https://github.com/kyleoconnell/gpu-accelerated-genomics).

288

289 **GCP Configuration**

290 Benchmarking on GCP leveraged virtual machines that were launched programmatically
291 for CPU machines, or manually for GPU machines. CPU workflows used the 'n2-standard-32'
292 machine type with Intel Xeon Cascade Lake processors with 32 vCPUs and 128 GB of RAM. We
293 assigned 1 TB of EBS storage to our instance. We launched these machines using a startup script
294 that installed the conda environment, then ran the snakemake workflows. All data was already
295 loaded on a machine image, and runtimes were concatenated from each snakemake rule using a
296 custom script. We also benchmarked the older generation E2 family of processors, but found the
297 run times to be much slower and thus only present the results for N2 processors here.

298 GPU benchmarking on GCP used the accelerator optimized a2-highgpu machine types with two
299 A100 GPUs, 24 vCPUs and 170GB RAM , four A100 GPUs with 48 vCPUs and 340 GB RAM,
300 and eight A100 GPUs with 96 vCPUs and 680 GB RAM. One virtual machine was utilized with
301 4 TB storage, which we stopped and resized between runs.

302

303 **AWS Configuration**

304 Benchmarking on AWS also used multiple virtual machines for CPU and GPU
305 benchmarking. CPU benchmarking used the C6i.8xlarge machine type, which has a 3rd
306 generation Intel Xeon Scalable processor with 32 vCPUs and 64 GiB RAM. We assigned 800
307 GB of EBS storage to our instance. We did some preliminary testing with the new ARM-based
308 processors but had issues with installing several of the dependencies (particularly with
309 mamba/conda), suggesting that a migration to ARM-based processors may prove problematic for
310 bioinformatics in the cloud.

311 We benchmarked two GPU machine families. First, we benchmarked the p4 machine
312 family which is similar to GCP a2-highgpu machines utilizing the latest NVIDIA A100 Tensor
313 Core GPUs with 8 GPUs with 96 vCPUs and 1152 GiB RAM. AWS currently only has one
314 machine type with A100 GPUs, the p4d.24xlarge, which only runs with 8 GPUs. To ensure
315 consistency with GCP, we ran the 8 GPU machine, but specified the number of GPUs to use in
316 our Parabricks commands for the smaller numbers of GPU runs. Because this machine type was
317 not compatible with the marketplace image (see below) we installed Parabricks manually using
318 scripts provided by NVIDIA. When possible (--cpu flag available) we limited the number of
319 CPUs available with the p4 machine, but some analysis used more CPUs on AWS than on GCP.

320 To compare GPU and CPU configurations directly with GCP, we further benchmarked
321 the p3 machine family using the ‘NVIDIA Clara Parabricks Pipelines’ AWS Marketplace image.
322 At the time of writing the image supported V100 GPUs (but not A100 GPUs), which are an older
323 model of Tensor Core GPU, on machine types p3.8xlarge with 4 GPUs and p3dn.24xlarge with 8
324 GPUs. The Marketplace image also had Parabricks preinstalled at a cost of \$0.30 for the
325 software. This configuration allowed us to directly compare 4, and 8 GPU machines with equal
326 CPU numbers between AWS and GCP. Again, we limited the number of CPUs available to the 2
327 GPU runs when possible. After we finished our analyses, NVIDIA wrote a helpful somatic
328 benchmarking guide (<https://github.com/clara-parabricks/NVIDIA-Clara-Parabricks-Somatic->
329 [Variant-Calling-AWS-Blog](#)).

330

331 **DGX Configuration**

332 We also conducted GPU benchmarking on an NVIDIA DGX Cluster (DGX SuperPOD), which
333 is a computing cluster with six DGX A100s, each of which contains eight NVIDIA A100 GPUs.
334 Although the cluster technically has 48 A100 GPUs available, Parabricks is only able to run on a
335 single DGX A100 system, thus limiting any Parabricks analyses to 8 GPUs. Jobs were launched
336 using a Kubernetes-based scheduler, allocating a max memory of 300 GB, and matching the
337 GPU and CPU configurations of the GCP/AWS runs, with the exception of GATK
338 HaplotypeCaller. For this workflow, we benchmarked times for 8 GPUs using 24, 48, 96, and
339 124 CPUs.

340

341 **DECLARATIONS**

342

343 **Availability of Data and Materials**

344 The dataset(s) supporting the conclusions of this article is(are) available in the GitHub at
345 <https://github.com/kyleoconnell/gpu-acclerated-genomics>.

346

347 **Author's Contributions**

348 KAO, CJL, TBC, DM, VRB, and JAK conceived the study. KAO, ZBY, RAC, and CJL
349 designed the study. KAO, ZBY, RAC, and CJL ran cloud-based analyses. KAO and JJC ran
350 DGX analyses. KAO, ZBY and HTE wrote the manuscript, and all authors read and approved of
351 the text.

352

353 **Competing Interests**

354 Deloitte Consulting LLP. and NVIDIA are alliance partners.

355

356 **REFERENCES**

357 Augustyn, D. R., Wycislik, Ł., & Mrozek, D. (2021). Perspectives of using Cloud computing in
358 integrative analysis of multi-omics data. *Briefings in Functional Genomics*, 20(4), 198–
359 206. <https://doi.org/10.1093/bfgp/elab007>

360 *Benchmarking NVIDIA Clara Parabricks Somatic Variant Calling Pipeline on AWS*. (2022, April
361 20). Amazon Web Services. <https://aws.amazon.com/blogs/hpc/benchmarking-nvidia->
362 [clara-parabricks-somatic-variant-calling-pipeline-on-aws/](https://aws.amazon.com/blogs/hpc/clara-parabricks-somatic-variant-calling-pipeline-on-aws/)

363 *Benchmarking NVIDIA Clara Parabricks Somatic Variant Calling Pipeline on AWS*. (2022, May
364 10). HPCwire. https://www.hpcwire.com/solution_content/aws/benchmarking-nvidia-
365 [clara-parabricks-somatic-variant-calling-pipeline-on-aws/](https://www.hpcwire.com/solution_content/aws/benchmarking-nvidia-clara-parabricks-somatic-variant-calling-pipeline-on-aws/)

366 *Benchmarking the NVIDIA Clara Parabricks germline pipeline on AWS*. (2021, November 23).
367 Amazon Web Services. <https://aws.amazon.com/blogs/hpc/benchmarking-the-nvidia->
368 [clara-parabricks-germline-pipeline-on-aws/](https://aws.amazon.com/blogs/hpc/clara-parabricks-germline-pipeline-on-aws/)

369 Cole, B. S., & Moore, J. H. (2018). Eleven quick tips for architecting biomedical informatics
370 workflows with cloud computing. *PLOS Computational Biology*, 14(3), e1005994.
371 <https://doi.org/10.1371/journal.pcbi.1005994>

372 Crowgey, E. L., Vats, P., Franke, K., Burnett, G., Sethia, A., Harkins, T., & Druley, T. E. (2021).

373 Enhanced processing of genomic sequencing data for pediatric cancers: GPUs and
374 machine learning techniques for variant detection. *Cancer Research*,
375 81(13_Supplement), 165. <https://doi.org/10.1158/1538-7445.AM2021-165>

376 Franke, K. R., & Crowgey, E. L. (2020). Accelerating next generation sequencing data analysis:
377 An evaluation of optimized best practices for Genome Analysis Toolkit algorithms.
378 *Genomics & Informatics*, 18(1), e10. <https://doi.org/10.5808/GI.2020.18.1.e10>

379 Gardy, J. L., & Loman, N. J. (2018). Towards a genomics-informed, real-time, global pathogen
380 surveillance system. *Nature Reviews Genetics*, 19(1), 9–20.
381 <https://doi.org/10.1038/nrg.2017.88>

382 Grossman, R. L. (2019). Data Lakes, Clouds, and Commons: A Review of Platforms for
383 Analyzing and Sharing Genomic Data. *Trends in Genetics: TIG*, 35(3), 223–234.
384 <https://doi.org/10.1016/j.tig.2018.12.006>

385 Grzesik, P., Augustyn, D. R., Wyciślik, Ł., & Mrozek, D. (2021). Serverless computing in omics
386 data analysis and integration. *Briefings in Bioinformatics*, bbab349.
387 <https://doi.org/10.1093/bib/bbab349>

388 Hawari, M. A., Hong, C. S., & Biesecker, L. G. (2021). SomatoSim: Precision simulation of
389 somatic single nucleotide variants. *BMC Bioinformatics*, 22, 109.
390 <https://doi.org/10.1186/s12859-021-04024-8>

391 Koppad, S., B, A., Gkoutos, G. V., & Acharjee, A. (2021). Cloud Computing Enabled Big Multi-
392 Omics Data Analytics. *Bioinformatics and Biology Insights*, 15, 11779322211035920.
393 <https://doi.org/10.1177/11779322211035921>

394 Krissaane, I., De Niz, C., Gutiérrez-Sacristán, A., Korodi, G., Ede, N., Kumar, R., Lyons, J.,
395 Manrai, A., Patel, C., Kohane, I., & Avillach, P. (2020). Scalability and cost-effectiveness
396 analysis of whole genome-wide association studies on Google Cloud Platform and
397 Amazon Web Services. *Journal of the American Medical Informatics Association:*
398 *JAMIA*, 27(9), 1425–1430. <https://doi.org/10.1093/jamia/ocaa068>

399 Krusche, P., Trigg, L., Boutros, P. C., Mason, C. E., De La Vega, F. M., Moore, B. L., Gonzalez-
400 Porta, M., Eberle, M. A., Tezak, Z., Lababidi, S., Truty, R., Asimenos, G., Funke, B.,
401 Fleharty, M., Chapman, B. A., Salit, M., Zook, J. M., & Global Alliance for Genomics and
402 Health Benchmarking Team. (2019). Best practices for benchmarking germline small-
403 variant calls in human genomes. *Nature Biotechnology*, 37(5), 555–560.
404 <https://doi.org/10.1038/s41587-019-0054-x>

405 Langmead, B., & Nellore, A. (2018). Cloud computing as a platform for genomic data analysis
406 and collaboration. *Nature Reviews. Genetics*, 19(4), 208–219.
407 <https://doi.org/10.1038/nrg.2017.113>

408 Larson, D. E., Harris, C. C., Chen, K., Koboldt, D. C., Abbott, T. E., Dooling, D. J., Ley, T. J.,
409 Mardis, E. R., Wilson, R. K., & Ding, L. (2012). SomaticSniper: Identification of somatic
410 point mutations in whole genome sequencing data. *Bioinformatics*, 28(3), 311–317.
411 <https://doi.org/10.1093/bioinformatics/btr665>

412 Leonard, C., Wood, S., Holmes, O., Waddell, N., Gorse, D., Hansen, D. P., & Pearson, J. V.
413 (2019). Running Genomic Analyses in the Cloud. *Studies in Health Technology and
414 Informatics*, 266, 149–155. <https://doi.org/10.3233/SHTI190787>

415 Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G.,
416 Durbin, R., & 1000 Genome Project Data Processing Subgroup. (2009). The Sequence
417 Alignment/Map format and SAMtools. *Bioinformatics*, 25(16), 2078–2079.
418 <https://doi.org/10.1093/bioinformatics/btp352>

419 Liu, B., Madduri, R. K., Sotomayor, B., Chard, K., Lacinski, L., Dave, U. J., Li, J., Liu, C., &
420 Foster, I. T. (2014). Cloud-based bioinformatics workflow platform for large-scale next-
421 generation sequencing analyses. *Journal of Biomedical Informatics*, 49, 119–133.
422 <https://doi.org/10.1016/j.jbi.2014.01.005>

423 Mölder, F., Jablonski, K. P., Letcher, B., Hall, M. B., Tomkins-Tinch, C. H., Sochat, V., Forster,
424 J., Lee, S., Twardziok, S. O., Kanitz, A., Wilm, A., Holtgrewe, M., Rahmann, S.,

425 Nahnsen, S., & Köster, J. (2021). Sustainable data analysis with Snakemake.

426 *F1000Research*, 10, 33. <https://doi.org/10.12688/f1000research.29032.2>

427 Nwadiugwu, M. C., & Monteiro, N. (2022). Applied genomics for identification of virulent

428 biothreats and for disease outbreak surveillance. *Postgraduate Medical Journal*.

429 <https://doi.org/10.1136/postgradmedj-2021-139916>

430 Poplin, R., Chang, P.-C., Alexander, D., Schwartz, S., Colthurst, T., Ku, A., Newburger, D.,

431 Dijamco, J., Nguyen, N., Afshar, P. T., Gross, S. S., Dorfman, L., McLean, C. Y., &

432 DePristo, M. A. (2018). A universal SNP and small-indel variant caller using deep neural

433 networks. *Nature Biotechnology*, 36(10), 983–987. <https://doi.org/10.1038/nbt.4235>

434 Poplin, R., Ruano-Rubio, V., DePristo, M. A., Fennell, T. J., Carneiro, M. O., Auwera, G. A. V.

435 der, Kling, D. E., Gauthier, L. D., Levy-Moonshine, A., Roazen, D., Shakir, K., Thibault,

436 J., Chandran, S., Whelan, C., Lek, M., Gabriel, S., Daly, M. J., Neale, B., MacArthur, D.

437 G., & Banks, E. (2018). *Scaling accurate genetic variant discovery to tens of thousands*

438 *of samples* (p. 201178). bioRxiv. <https://doi.org/10.1101/201178>

439 Ray, U., Krishnan, V., Bahmani, A., Pan, C., Bettinger, K., Tsao, P., Mueller, F., & Snyder, M.

440 (2021). Hummingbird: Efficient Performance Prediction for Executing Genomics

441 Applications in the Cloud. *Bioinformatics*, 37(17), 2537–2543.

442 Rosati, S. (2020). Comparison of CPU and Parabricks GPU Enabled Bioinformatics Software for

443 High Throughput Clinical Genomic Applications. *Master's Thesis (2009 -)*, 43.

444 Tanjo, T., Kawai, Y., Tokunaga, K., Ogasawara, O., & Nagasaki, M. (2021). Practical guide for

445 managing large-scale human genome data in research. *Journal of Human Genetics*,

446 66(1), 39–52. <https://doi.org/10.1038/s10038-020-00862-1>

447 Van der Auwera, G. A., & O'Connor, B. D. (2020). *Genomics in the cloud: Using Docker, GATK,*

448 *and WDL in Terra* (1st ed.). O'Reilly Media.

449 Zhang, Q., Liu, H., & Bu, F. (2021). *High performance of a GPU-accelerated variant calling tool*

450 *in genome data analysis* [Preprint]. Bioinformatics.

451 <https://doi.org/10.1101/2021.12.12.472266>

452 Zhao, S., Agafonov, O., Azab, A., Stokowy, T., & Hovig, E. (2020). *Accuracy and efficiency of*
453 *germline variant calling pipelines for human genome data* (p. 2020.03.27.011767).

454 bioRxiv. <https://doi.org/10.1101/2020.03.27.011767>

455 Zook, J. M., Catoe, D., McDaniel, J., Vang, L., Spies, N., Sidow, A., Weng, Z., Liu, Y., Mason,
456 C. E., Alexander, N., Henaff, E., McIntyre, A. B. R., Chandramohan, D., Chen, F.,
457 Jaeger, E., Moshrefi, A., Pham, K., Stedman, W., Liang, T., ... Salit, M. (2016).
458 Extensive sequencing of seven human genomes to characterize benchmark reference
459 materials. *Scientific Data*, 3(1), 160025. <https://doi.org/10.1038/sdata.2016.25>

460

461 Table 1: Results of benchmarking for AWS, GCP and NVIDIA DGX workflow runs. AWS results presented here
462 are for the p3 family with the NVIDIA Tesla V100 GPU, results for the p4 family with the A100 GPU are shown in
463 Table S1.

Platform	Pipeline	VM-Type	Variant-Caller	Time (min)	Time (hours)	Cost (\$)	Fold Acceleration	% Cost-Savings
AWS	Germline	C6i.8xlarge	DeepVariant	1317.3	21.96	29.9	—	—
		GPU2		145.16	2.42	29.61	9.07	0.83
		GPU4		97.07	1.62	19.80	13.57	33.68
		GPU8		42.19	0.7	21.95	31.22	26.49
GCP		n2-32		1128	18.8	32.9	—	—
		GPU2		156	2.6	19.4	7.2	41.03
		GPU4		72	1.2	18.3	15.7	44.38
		GPU8		42.6	0.71	20.9	26.5	36.47
DGX		GPU2		87.9	1.47	—	—	—
		GPU4		49.1	0.82	—	—	—
		GPU8		27.05	0.45	—	—	—

Platform	Pipeline	VM-Type	Variant-Caller	Time (min)	Time (hours)	Cost (\$)	Fold Acceleration	% Cost-Savings
AWS	Germline	C6i.8xlarge	HaplotypeCaller	2175.9	36.26	49.32	—	—
		GPU2		131.99	2.2	26.93	16.49	45.41
		GPU4		88.27	1.47	18	24.65	63.49
		GPU8		41.51	0.69	21.60	52.42	56.21
GCP		n2-32		2328	38.8	67.9	—	—
		GPU2		118.8	1.98	13.5	19.6	80.12
		GPU4		57.6	0.96	14.1	40	79.23
		GPU8		35.4	0.59	17.5	65.8	74.23
DGX		GPU2		64.6	1.08	—	—	—
		GPU4		39	0.65	—	—	—
		GPU8		24.4	0.41	—	—	—
AWS	Somatic	C6i.8xlarge	LoFreq	180.2	3	4.1	—	—
		GPU2		145.14	2.42	29.61	1.24	-625.07
		GPU4		109.23	1.82	22.28	1.65	-445.68
		GPU8		57.18	0.95	29.75	3.15	-628.55
GCP		N2-32		277.8	4.63	8.1	—	—
		GPU2		155.2	2.59	19	1.8	-134.5
		GPU4		110.9	1.85	27.1	2.5	-235
		GPU8		61.4	1.02	30.1	4.5	-271
DGX		GPU2		113.71	1.9	—	—	—
		GPU4		70.41	1.18	—	—	—
		GPU8		49.5	0.83	—	—	—

Platform	Pipeline	VM-Type	Variant-Caller	Time (min)	Time (hours)	Cost (\$)	Fold Acceleration	% Cost-Savings
AWS	Somatic	C6i.8xlarge	Muse	425.1	7.09	9.6	—	—
		GPU2		65.17	1.09	13.29	6.52	-37.97
		GPU4		61.35	1.02	12.52	6.93	-29.88
		GPU8		22.27	0.37	11.59	19.09	-20.23
GCP		N2_32		621.8	10.36	18.1	—	—
		GPU2		44.2	0.74	5.4	14.1	70.1
		GPU4		32.4	0.54	7.9	19.2	56.2
		GPU8		28.5	0.48	14	21.8	22.9
DGX		GPU2		36	0.6	—	—	—
		GPU4		23.84	0.4	—	—	—
		GPU8		22.7	0.38	—	—	—
AWS	Somatic	C6i.8xlarge	Mutect2	414.51	6.91	9.40	—	—
		GPU2		28.4	0.47	5.79	14.60	38.34
		GPU4		21.54	0.36	4.39	19.24	53.23
		GPU8		28.6	0.48	14.88	14.50	-58.36
GCP		N2_32		487.7	8.13	14.2	—	—
		GPU2		32.9	0.55	4.03	14.8	71.63
		GPU4		16.7	0.28	4.1	29.3	71.29
		GPU8		31	0.52	15.2	15.7	-7.06
DGX		GPU2		19.17	0.32	—	—	—
		GPU4		17.2	0.29	—	—	—
		GPU8		23.4	0.39	—	—	—

Platform	Pipeline	VM-Type	Variant-Caller	Time (min)	Time (hours)	Cost (\$)	Fold Acceleration	% Cost-Savings
AWS	Somatic	C6i.8xlarge	SomaticSniper	391.9	6.53	8.88	—	—
		GPU2		83.7	1.4	17.07	4.68	-92.28
		GPU4		134.12	2.24	27.36	2.92	-208.11
		GPU8		144.48	2.41	75.17	2.71	-746.54
GCP		N2_32		482.8	8.05	14.1	—	—
		GPU2		84.8	1.41	10.4	5.7	26.18
		GPU4		69.1	1.15	16.9	7	-20.33
		GPU8		100.5	1.68	49.3	4.8	-250.2
DGX		GPU2		77.54	1.29	—	—	—
		GPU4		65	1.08	—	—	—
		GPU8		63.5	1.06	—	—	—

464

465

466 FIGURE CAPTIONS

467 Figure 1: Comparison of execution times of variant calling algorithms on CPU and GPU
 468 environments between AWS and GCP. A 32 vCPU machine with the latest processors was used
 469 for CPU benchmarking on both cloud platforms. Here we show results for varying numbers of
 470 NVIDIA Tesla V100 GPUs running the Parabricks bioinformatics suite for AWS, and NVIDIA
 471 Tesla A100 GPUs for GCP.

472 Figure 2. GPU benchmarking results for NVIDIA Tesla GPUs. On GCP and the DGX results are
 473 shown for A100 GPUs, whereas AWS results are shown for the V100 GPU runs.

474 Figure 3: Comparison of runtimes between V100 and A100 GPU machines on AWS

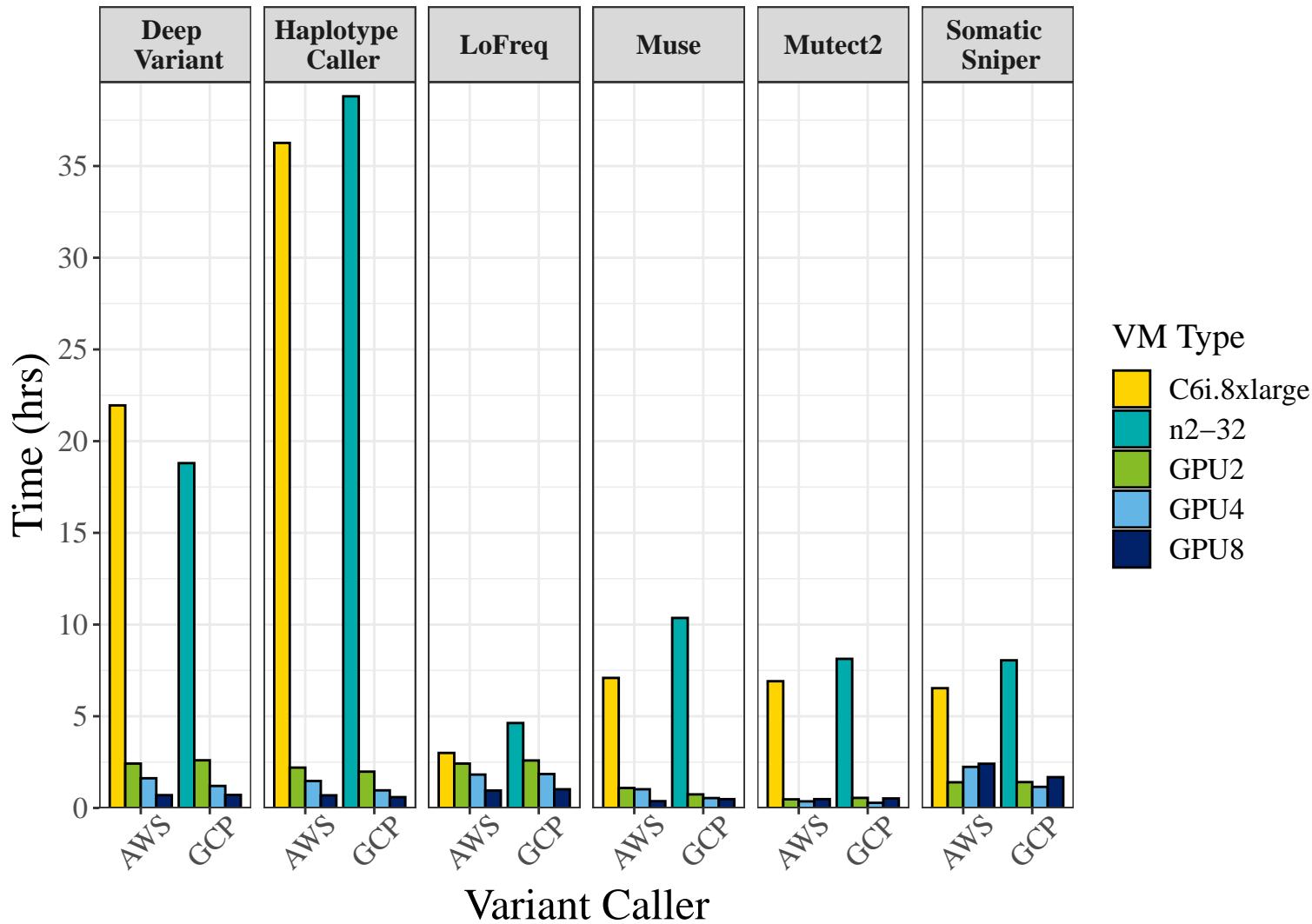
475 Figure 4. Comparison of AWS (V100 GPU machine) vs. GCP GPU cost savings per variant
 476 caller. Percentage of total cost savings shows a majority of higher cost savings using GPUs in
 477 algorithms optimized for GPU-acceleration, but losses when algorithms are not well optimized

478 Figure 5. Comparison of AWS V100 vs. GCP A100 GPU cost ratio per variant caller. Cost ratio
 479 being the ratio between cost per hour and fold speed-up. Cost per fold-speedup shows the benefit

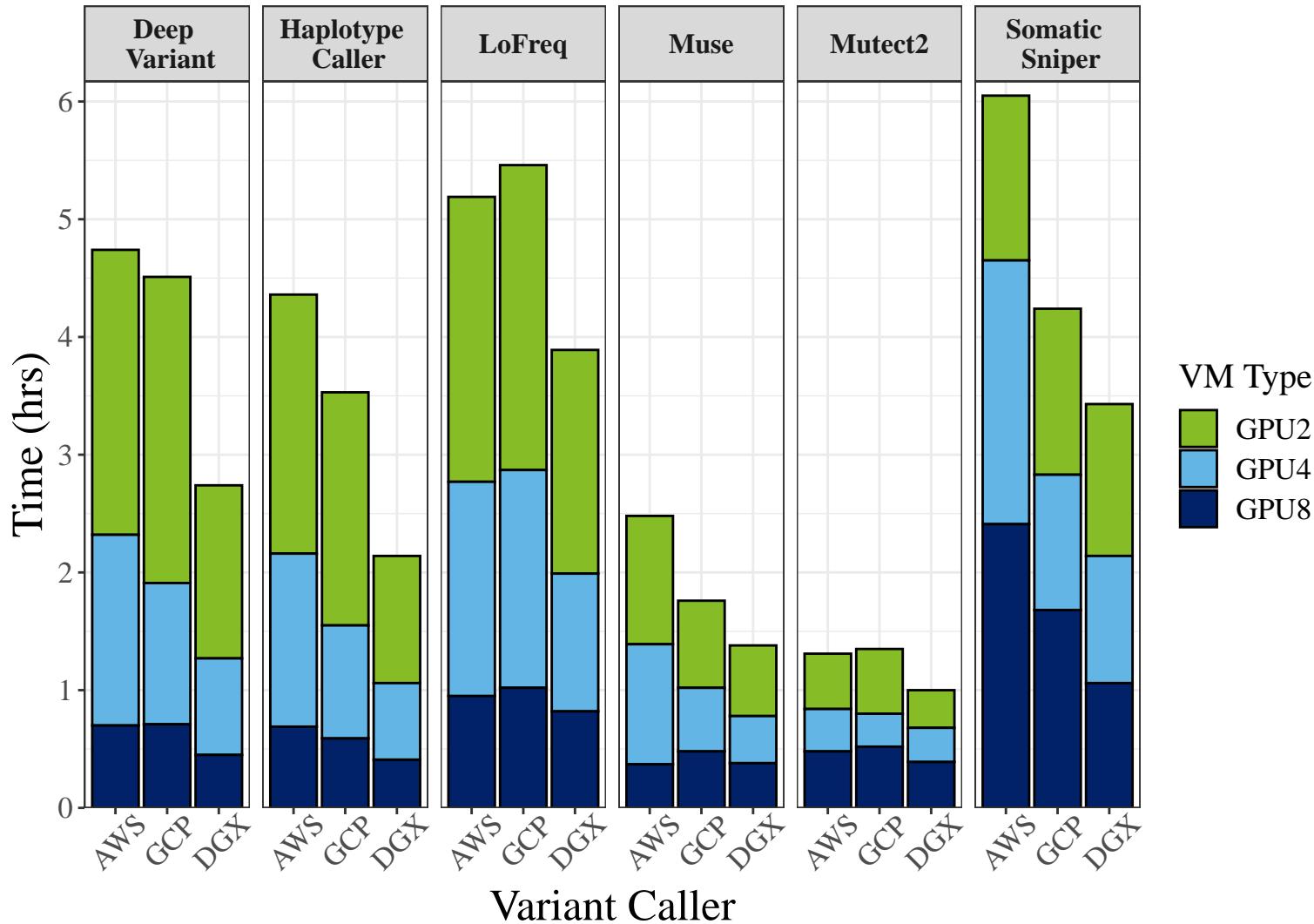
480 of harnessing GPU over CPU in select algorithms, while other algorithms are more cost-efficient
481 with CPUs.

482

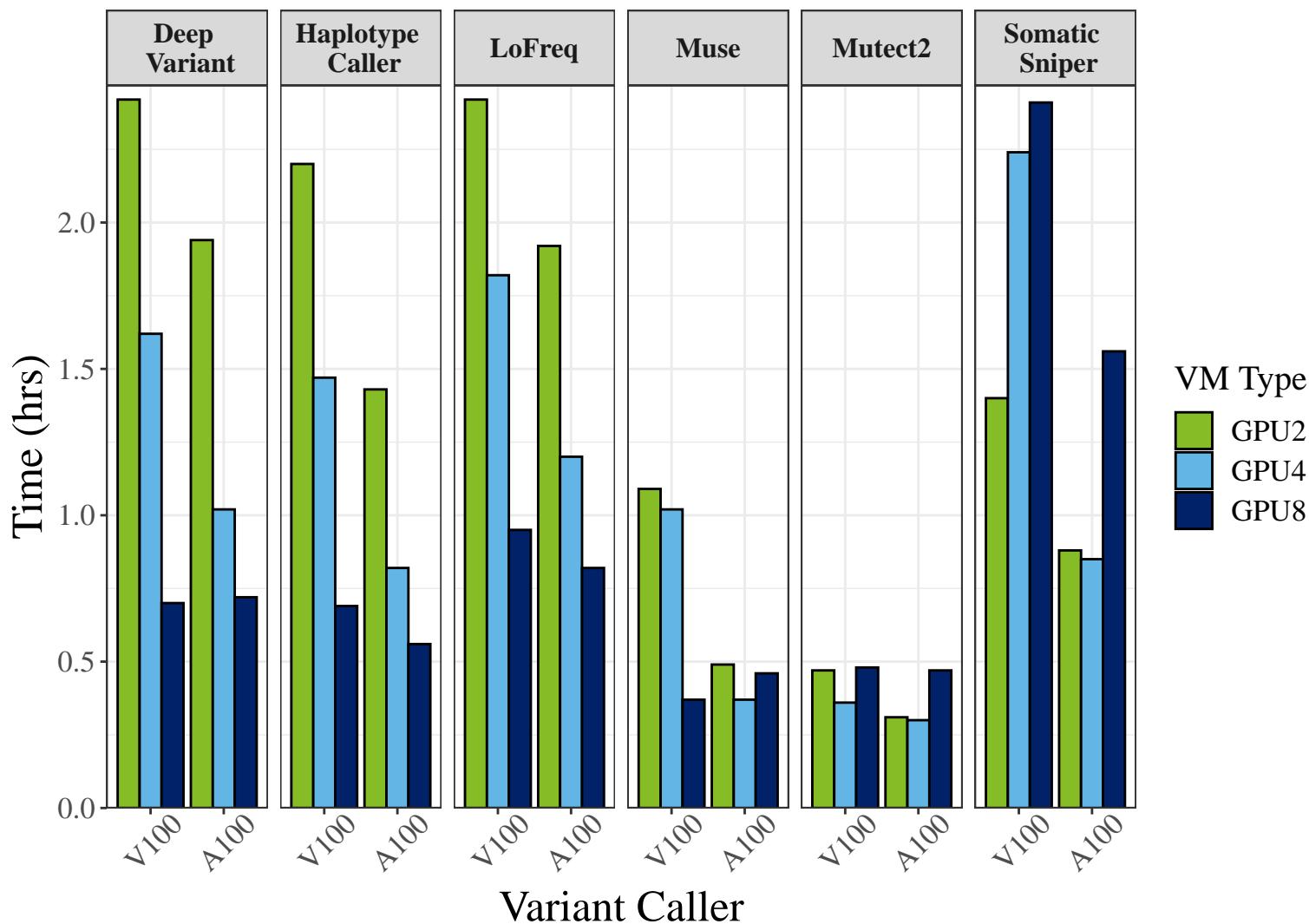
Runtimes of All Cloud-Based Analyses



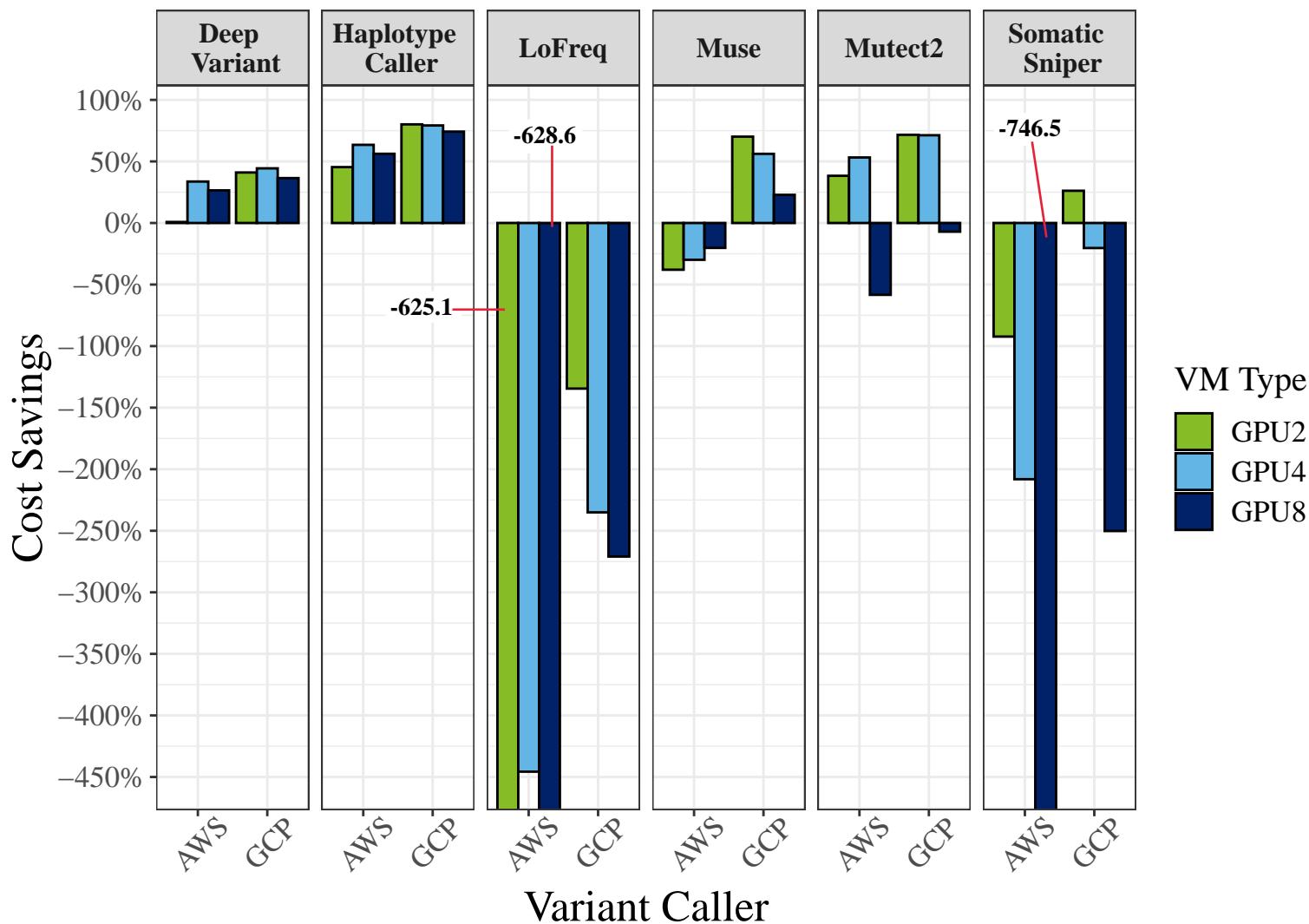
GPU Runtimes Across Platforms



AWS GPU Hours per Workflow



GPU Cost Savings per Workflow



GPU Cost per Fold–Speedup

