Scaling computational genomics to millions of individuals with GPUs

Amaro Taylor-Weiner*[1,2], François Aguet*[1], Nicholas J. Haradhvala[1], Sager Gosai[1,2], Jaegil Kim[1], Kristin Ardlie[1], Eliezer M. Van Allen[1,3,5], and Gad Getz[#1,4,6]

1 Broad Institute of Harvard and MIT, Cambridge, MA, USA
2 Harvard University, Cambridge, MA, USA
3 Department of Medical Oncology, Dana-Farber Cancer Institute, Boston, MA USA
4 Department of Pathology, Harvard Medical School, Boston, MA, USA
5 Department of Medicine, Harvard Medical School, Boston, MA, USA
6 Department of Pathology, Massachusetts General Hospital, Boston, MA, USA
* these authors contributed equally
# Address correspondence to: Gad Getz gadgetz@broadinstitute.org

To the Editor:

Current genomics methods and pipelines were designed to handle tens to thousands of samples, but will soon need to scale to millions to keep up with the pace of data and hypothesis generation in biomedical science. The computational costs associated with processing these growing datasets will become prohibitive without improving the computational efficiency and scalability of methods. For example, methods in population genetics, such as genome-wide association studies (GWAS) or mapping of quantitative trait loci (QTL), involve billions of regressions between genotypes and phenotypes. Currently, the state-of-the-art infrastructure for performing these tasks are large-scale clusters of central processing units (CPUs), often with thousands of cores that result in significant cost[1] (960 cores on a standard Google Cloud machine costs $7,660.80 per day of compute). In contrast to CPUs, which feature relatively few cores (i.e. Intel's i9 has 6 cores), a single graphical processing unit (GPU) contains thousands of cores (i.e. Nvidia's P100 has 3,584 cores). GPUs are optimized for computing highly parallelizable mathematical operations, and use of GPUs has revolutionized the field of machine learning, where training extremely deep artificial neural networks requires performing billions of identical operations[2].

Here, we hypothesized that implementation of population genetics methods using recently developed machine-learning libraries for GPUs[3,4] would result in significant improvements to computational efficiency and enable scaling to hundreds of thousands of patient samples. Previous work has shown the benefits of using GPUs to scale bioinformatics methods, but existing implementations are often complex and based on specialized libraries, limiting their extendibility and adoption[5–7]. To examine the differences in efficiency and benchmark the use of CPUs and GPUs for large-scale genomic analyses, we re-implemented methods for two commonly performed computational genomics tasks in a GPU environment: (i) QTL mapping[8,9] (which we call tensorQTL) and Bayesian non-negative matrix factorization[10] (named SignatureAnalyzer-GPU). We implemented both methods using the open-source machine-learning framework TensorFlow[3] for Python.

To study the impact in cost and run-time of these GPU implementations, we executed the same exact scripts in identical environments (configured with and without a GPU), and also compared them to previous CPU-based implementations. For SignatureAnalyzer-GPU (SA-GPU), we used the mutation counts matrix generated by the Pan-Cancer Analysis of Whole Genomes (PCAWG) Consortium, which contains 2,624 tumors represented by 1697 mutational features of somatic single nucleotide variants and short insertions and deletions (defined based on their sequence contexts)[11]. Our GPU implementation ran approximately 200 times faster than the current implementation of SignatureAnalyzer (SA) in R with mean times for 10,000 iterations of 194.8 min using SA vs. 1.09 min using SA-GPU (Fig. 1a). Using simulated data, we showed that SA-GPU scales linearly with the number of samples (Supplementary Fig. 1a). When applied to previously published mutational signatures generated by SA[12], we found the results of the two methods were essentially identical (mean $R^2 = 0.994$, min $R^2 = 0.96$; Fig. 1b, an exact match is not expected due to the stochastic nature of the underlying algorithm).

Similarly, for tensorQTL benchmarking, we generated random data representing up to 50,000 people, each with 10^7 genotypes representing common variants. For each individual, we also simulated up to 50,000 phenotypes, resulting in 500×10^9 all-against-all association tests (each calculated over up to 50,000 individuals). Our implementation enabled *cis*-QTL mapping, with permutations to estimate empirical false discovery rate, >250x times faster than the current state-of-the-art implementation (FastQTL[9]; Fig 1c). Likewise, *trans*-QTL mapping (i.e., 500 billion regressions) took less than 10 minutes (a ~200x increase in speed compared to running without a GPU, Fig. 1c & Supplementary Fig. 1b). Our current implementation does not scale linearly as a function of samples (Supplementary Fig. 1c) due to limitations in data transfer from the memory of the CPU to the GPU rather than computational capacity; we leave this additional optimization for future work (Fig. 1d, Supplementary Fig. 1c). At the time of writing, GPU compute time could be purchased for ~$0.5/hour on multiple cloud platforms compared to ~$0.01-0.05/hour for CPUs, but despite the higher GPU cost, these analyses are ~5-10 fold cheaper on GPUs due to the vastly shorter runtimes. We used data from the V6p and V7 releases of GTEx[13] generated using Matrix eQTL[8] and FastQTL[9], respectively, to demonstrate the reproducibility of our implementation (Fig. 1e & 1f).

Overall, implementation of many commonly used methods in genomics on GPUs can result in a significant increase in speed and reduction of costs compared to current approaches. These findings indicate that the scale of computations made possible with GPUs will enable investigation of previously unanswerable hypotheses involving more complex models, larger datasets, and more accurate empirical measurements. For example, the application of Bayesian NMF to a million single cells takes approximately 6 hours using our GPU-based approach compared to approximately 50 days with CPUs. Similarly, our GPU implementation enables computation of empirical p-values for *trans*-QTL, which is intractable on CPUs. Importantly, the reduced costs and wide availability of GPUs will also enable more

researchers around the world to perform these complex analyses. Based on our results and the availability of open-source frameworks like TensorFlow[3] or PyTorch[4], which greatly facilitate development of mathematical methods for GPU environments without need for specialized knowledge of GPU architecture, we recommend a transition to GPU-based computing for a wide range of computational genomics methods.

Software Availability: All software is available on Github and implemented in python using open source libraries.
TensorQTL : https://github.com/broadinstitute/tensorqtl
SignatureAnalyzer-GPU: https://github.com/broadinstitute/signature_analyzer_tf

Conflicts of interest: A.T.W holds stock in Nvidia and Google (The developers of TensorFlow). A.T.W, F.A and G.G. have filed a patent application related to this approach. G.G. receives research funds from IBM and Pharmacyclics. G.G is an inventor on multiple bioinformatics related patent applications. E.M.V. is a consultant for Tango Therapeutics, Genome Medical, Invitae, Foresite Capital, and Illumina. E.M.V. received research support from Novartis and BMS, as well as travel support from Roche/Genentech. E.M.V. is an equity holder of Syapse, Tango Therapeutics, Genome Medical. E.M.V holds stock in Microsoft.

Author Contributions: A.T.W., and F.A. outlined and planned development. A.T.W., F.A., G.G., E.M.V prepared and reviewed manuscript and figures. F.A. and A.T.W developed tensorQTL. A.T.W., N.H., J.K. and S.G. developed SignatureAnalyzer-GPU. F.A. performed and planned benchmarking analysis for tensorQTL. A.T.W., N.H., and J.K. performed and planned benchmarking analysis for SignatureAnalyzer-GPU. All authors reviewed and edited the manuscript.

1. Bycroft, C. *et al.* The UK Biobank resource with deep phenotyping and genomic data. *Nature* **562,** 203–209 (2018).
2. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521,** 436–444 (2015).
3. Abadi, M. *et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*.
4. Paszke, A. *et al.* Automatic differentiation in PyTorch. (2017).
5. McArt, D. G. *et al.* cudaMap: a GPU accelerated program for gene expression connectivity mapping. *BMC Bioinformatics* **14,** 305 (2013).
6. Mejía-Roa, E. *et al.* NMF-mGPU: non-negative matrix factorization on multi-GPU systems. *BMC Bioinformatics* **16,** 43 (2015).
7. Schatz, M. C., Trapnell, C., Delcher, A. L. & Varshney, A. High-throughput sequence alignment using Graphics Processing Units. *BMC Bioinformatics* **8,** 474 (2007).
8. Shabalin, A. A. Matrix eQTL: ultra fast eQTL analysis via large matrix operations. *Bioinformatics* **28,** 1353–8 (2012).
9. Ongen, H., Buil, A., Brown, A. A., Dermitzakis, E. T. & Delaneau, O. Fast and efficient QTL mapper for thousands of molecular phenotypes. *Bioinformatics* **32,** 1479–1485 (2016).
10. Kim, J. *et al.* Somatic ERCC2 mutations are associated with a distinct genomic signature in urothelial tumors. *Nat. Genet.* **48,** 600–606 (2016).
11. Alexandrov, L. *et al.* The Repertoire of Mutational Signatures in Human Cancer. *bioRxiv* 322859 (2018). doi:10.1101/322859
12. Haradhvala, N. J. *et al.* Distinct mutational signatures characterize concurrent loss of

polymerase proofreading and mismatch repair. *Nat. Commun.* **9,** 1746 (2018).
13. GTEx Consortium. Genetic effects on gene expression across human tissues. *Nature* **550,** 204–213 (2017).

Figure Legends:

Figure 1: Performance of GPU implementations for QTL mapping and signature analysis. a) Average run time to compute 10,000 iterations of Bayes NMF using SignatureAnalyzer (SA) in R (purple) and SignatureAnalyzer-GPU (SA-GPU; gold). b) Correlation heat map of mutation signatures derived from the R and GPU implementations of SignatureAnalyzer using the same input mutation counts matrix. c) Comparison of runtimes for *cis*-QTL (FastQTL on CPU; gold and tensorQTL on GPU; purple) and *trans*-QTL (tensorQTL on CPU and GPU). d) GPU runtime of tensorQTL for the indicated numbers of samples and phenotypes. e) Nominal significant *trans*-eQTL p-values from the V6p GTEx release[13] replicated using tensorQTL. e) Empirical *cis*-eQTL p-values from the V7 GTEx release replicated using tensorQTL. Error bars indicate standard deviation of the mean.

Supplementary Figure 1: GPU performance scaling of SignatureAnalyzer-GPU and tensorQTL. a) SignatureAnalyzer-GPU runtime scales linearly as a function of number of samples. b) GPU-to-CPU runtime ratio for tensorQTL, across the indicated phenotype and sample sizes, for 10^7 common variants. The ratio is non-constant due to data load and CPU-to-GPU memory input/ouput times ("i/o") that are more limiting for large sample sizes (number of individuals). c) CPU runtime of tensorQTL for the indicated range of sample and phenotype sizes (left panel). CPU runtimes scale linearly demonstrated by the collapse of the compute time when measured as a function of number of operations (approximated as phenotypes x samples x variants; middle panel), whereas GPU runtimes do not show this collapse for large sample sizes due to i/o limitations (right panel).

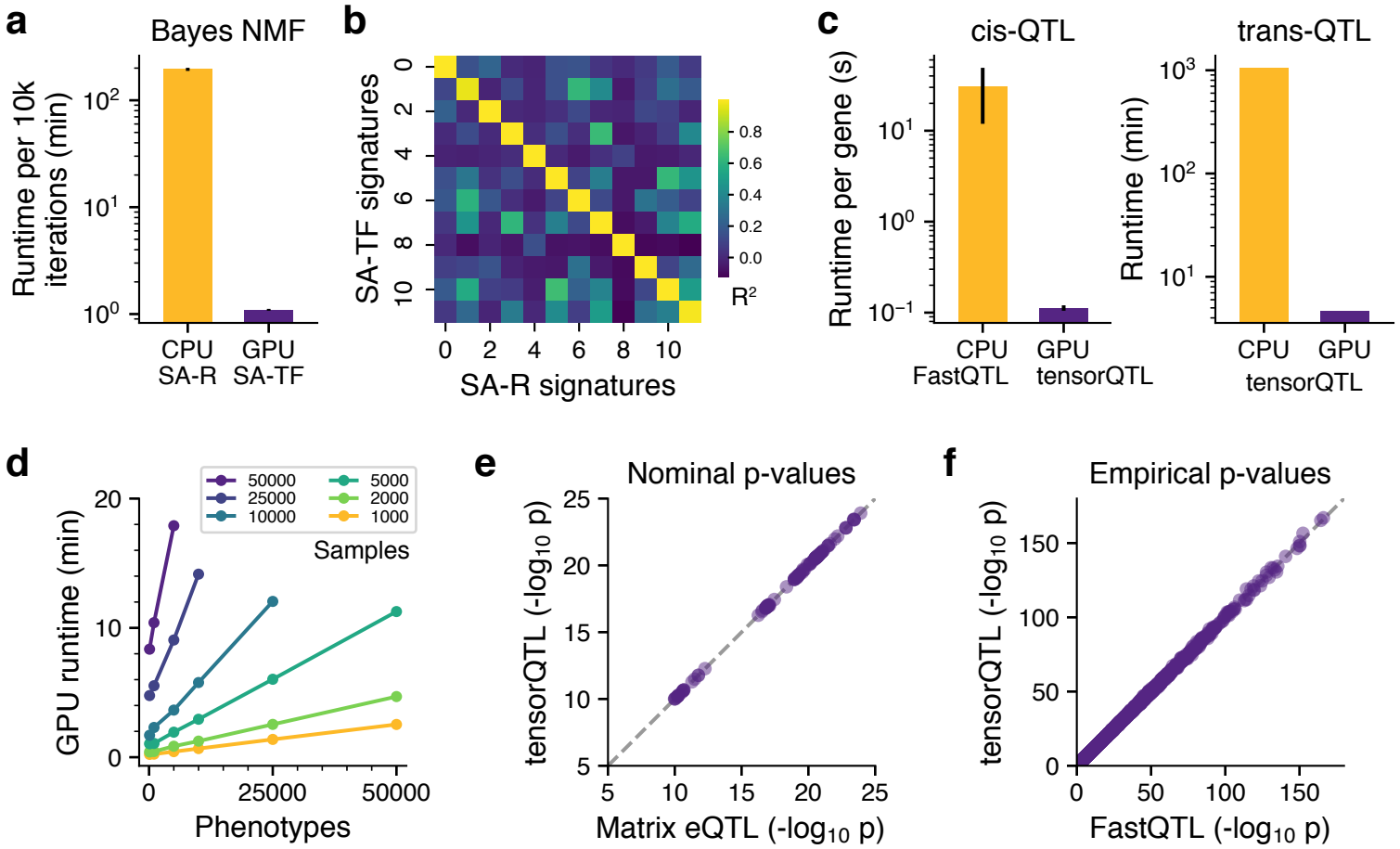
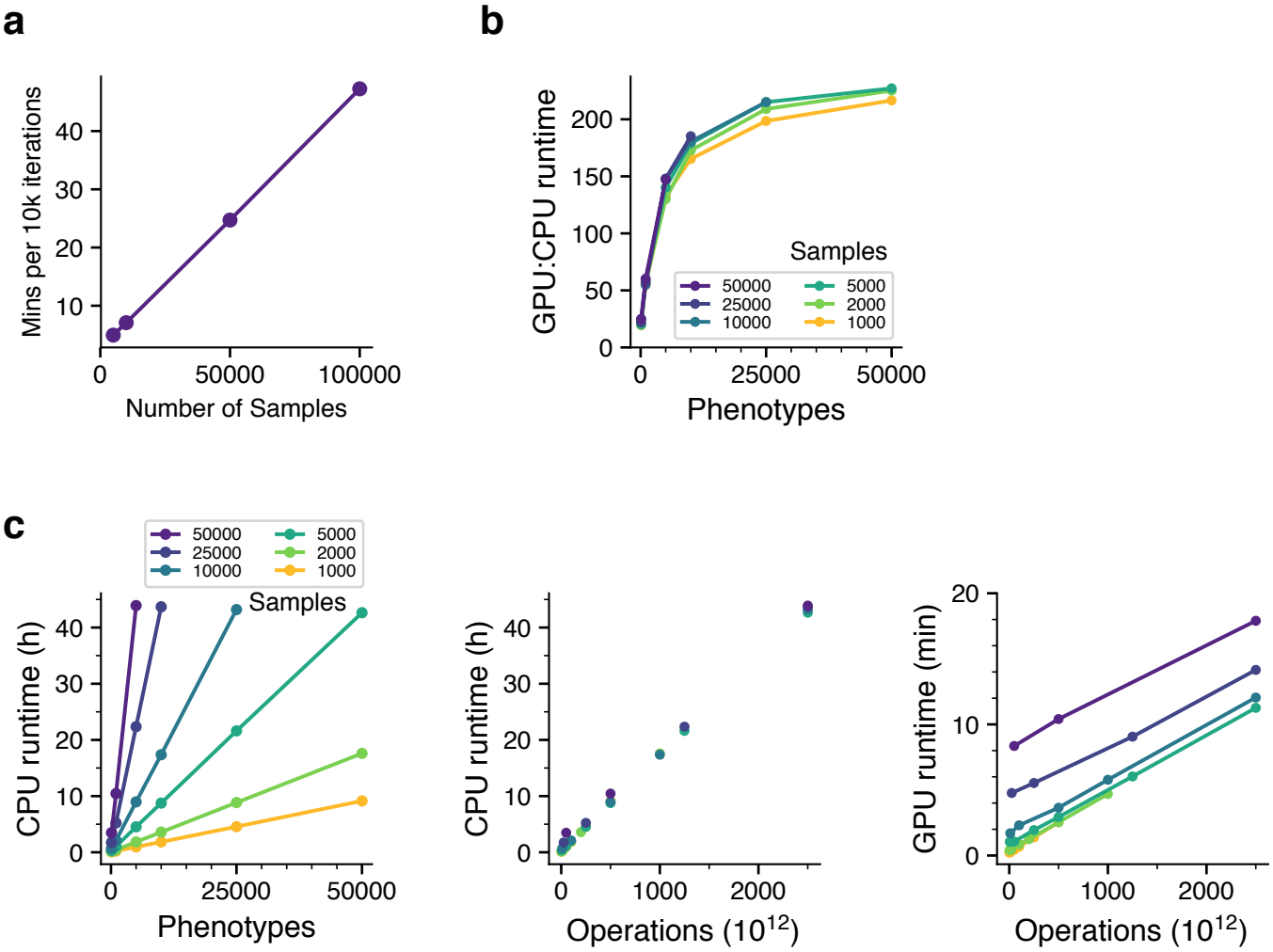

a
Bayes NMF
Runtime per 10k iterations (min)
CPU SA-R
GPU SA-TF
b
SA-TF signatures
SA-R signatures
R²
c
cis-QTL
trans-QTL
Runtime per gene (s)
Runtime (min)
CPU FastQTL
GPU tensorQTL
CPU tensorQTL
GPU tensorQTL
d
GPU runtime (min)
Phenotypes
Samples
50000
25000
10000
5000
2000
1000
e
Nominal p-values
tensorQTL (-log10 p)
Matrix eQTL (-log10 p)
f
Empirical p-values
tensorQTL (-log10 p)
FastQTL (-log10 p)

Figure 1



Supplementary Figure 1