Biology Does Not Run on One Chip

Precision, Memory and Energy Needs of Life-Science Computing across Six Data-Centre GPUs

Abhay Chakra Sadineni
University of California, Berkeley
abhaychakra@berkeley.edu

November 2026
DOI: 10.5281/zenodo.24210929

Abstract

Data-centre GPUs are now designed for training and serving large language models, and each generation's headline gain is in low-precision tensor arithmetic. Life-science computing is not one workload. Molecular dynamics runs in single precision with fixed-point accumulation. Electronic-structure codes need double precision. Structure prediction is limited by memory capacity. Biological foundation models look like any other large model. Genomics pipelines, cryo-electron microscopy and optical and acoustic simulation each have their own binding constraint. This paper sets public specifications for six NVIDIA data-centre GPUs from four architectures, from the 2017 V100 to the 2025 Blackwell Ultra, against the published hardware needs of eight classes of biological computation. Per watt, dense 16-bit tensor throughput (FP16 on the V100, BF16 since) rose 5.4 times from the V100 to the B200. Non-tensor single-precision throughput per watt peaked with the H100 and has since fallen 22 percent. Native double-precision throughput per watt also peaked with the H100, fell 61 percent on the B200, and fell to about 1 percent of the H100's level on Blackwell Ultra. The energy cost of a computation therefore depends on its precision as much as on the chip's age. At peak rates, 10^{18} double-precision operations cost about 2.9 kWh on an H100, 7.5 kWh on a B200 and roughly 280 kWh on native Blackwell Ultra, while 10^{18} BF16 operations cost 0.20 kWh on an H100 and 0.12 kWh on a B200. An AlphaFold 3 prediction of 5,120 tokens runs 1.8 times faster on an H100 than on an A100 but uses about the same GPU energy. The paper ends with a matching table that a research organisation can use to buy or rent hardware by workload rather than by generation.

Keywords: GPU architecture; numerical precision; molecular dynamics; protein structure prediction; biological foundation models; genomics; cryo-EM; energy.

1 Introduction

In August 2025 NVIDIA described Blackwell Ultra, its newest data-centre GPU, as offering 50 percent more four-bit floating-point throughput than its predecessor [1]. The same chip delivers about 1.4 trillion double-precision operations per second per package, against 37 to 40 on the B200 it followed, depending on the B200's power configuration [2, 3]. For a language-model operator that trade is sensible. For a computational chemist it removes most of the double-precision arithmetic the previous chip had.

That trade is the clearest sign of a larger shift. Accelerators are now shaped by the needs of one workload, and biology runs many. A lab that simulates proteins, predicts their structures, calls

variants from sequencing runs, reconstructs cryo-EM maps and trains a foundation model on all of it needs different things from its hardware for each job. Buying the newest chip for every job wastes money on some and fails outright on others.

This paper asks two questions. Which properties of a GPU actually limit each class of biological computation? And how have those properties, and the energy cost of using them, changed across the last four NVIDIA architectures? Section 2 sets out the four properties that matter. Section 3 tracks them across generations. Section 4 goes workload by workload. Section 5 turns the comparison into energy per unit of work, and Section 6 gives a matching table.

2 Four properties of a workload

Four properties decide which hardware suits a computation.

The first is numerical precision. Double precision (FP64) carries about 16 significant decimal digits and is the default in quantum chemistry and many solvers. Single precision (FP32) carries about 7. Tensor formats such as BF16, FP8 and FP4 carry fewer still and run on dedicated matrix units that deliver most of a modern GPU's quoted throughput.

The second is working-set memory. A job that does not fit in GPU memory must be split across devices or spill to host memory, at a large cost in speed.

The third is arithmetic intensity, the number of operations performed per byte moved to or from memory. The roofline model shows that a kernel whose intensity is below the chip's ratio of peak throughput to bandwidth, its ridge point, is limited by bandwidth and cannot use the extra arithmetic [4]. Peak arithmetic has grown faster than bandwidth for two decades, so more workloads sit below the ridge with each generation [5].

The fourth is parallel structure. Some jobs are one large calculation that must be spread across many GPUs with fast links. Others are thousands of small, independent jobs that fit on one modest GPU each.

3 Six GPUs from four architectures

Chip	Year	FP64 (TFLOPS)	FP32 (TFLOPS)	BF16 dense (TFLOPS)	Bandwidth (TB/s)	Memory (GB)	Power (W)
V100 SXM2	2017	7.8	15.7	125	0.90	32	300
A100 SXM	2020	19.5	19.5	312	2.04	80	400
H100 SXM	2022	67	67	989	3.35	80	700
H200 SXM	2024	67	67	989	4.80	141	700
B200 HGX	2025	37	75	2,250	7.70	180	1,000
B300 (GB300)	2025	1.39	83.3	2,500	8.00	288	1,400

Table 1. Per-GPU specifications. FP64 is the best native rate, using the FP64 tensor path where one exists. FP32 is the non-tensor rate. BF16 is dense, without structured sparsity (the V100 figure is FP16, as it has no BF16). Sources are NVIDIA datasheets and product pages for the V100, A100, H100, H200 and GB300 [6–10], the Lenovo HGX B200 product guide [3], and Tom's Hardware for the Blackwell Ultra power rating [2]. The B300 values are GB300 NVL72 rack figures divided by 72 GPUs, with dense BF16 taken as half the sparse rating [2, 10]. The B200 and B300 rows therefore use different board configurations, a 1,000 W HGX part and a 1,400 W rack part, and per-watt comparisons between them mix the two. The air-cooled HGX B300 runs at lower power and lower throughput.

Figure 1 divides each capability by the GPU's rated power. Four lines that rose together until 2022 have since separated.

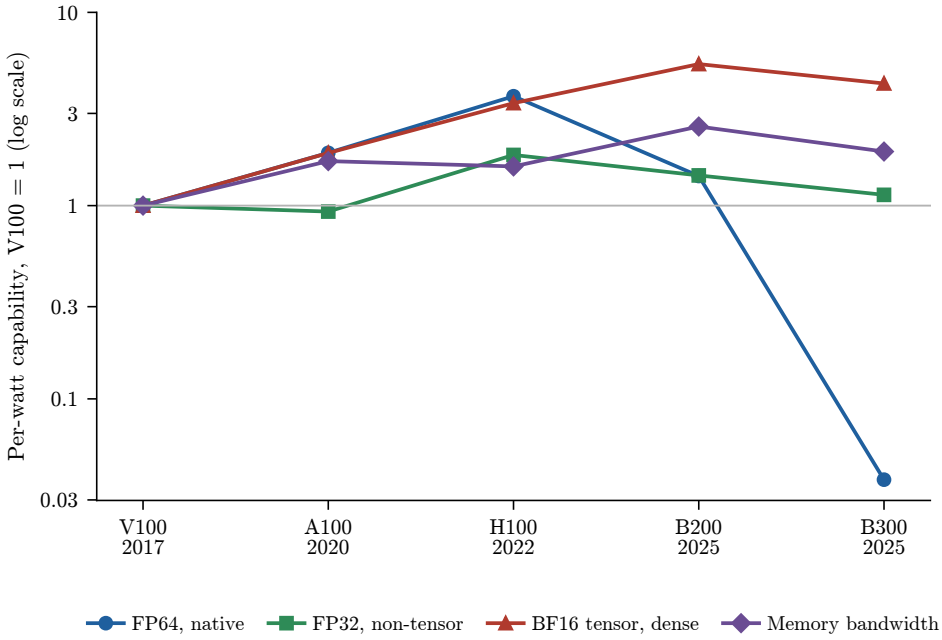


Figure 1. Capability per watt of GPU power, indexed to the V100, computed from Table 1. The B300 point uses the 1,400 W GB300 rating, which understates its per-watt figures relative to an air-cooled HGX B300.

Dense BF16 throughput per watt rose 5.4 times from the V100 to the B200. That is the gain the industry advertises, and it is real for workloads that use tensor units.

Non-tensor FP32 per watt barely moved from the V100 to the A100, nearly doubled on the H100, and has since fallen. The B200 delivers 22 percent less FP32 per watt than the H100, and the GB300 configuration 38 percent less.

Native FP64 per watt followed the same path more steeply. It rose 3.7 times to the H100, fell 61 percent on the B200, and on Blackwell Ultra fell to about 1 percent of the H100's level.

Memory bandwidth per watt rose 2.6 times to the B200, less than half the gain in BF16. The BF16 ridge point roughly doubled, from 153 operations per byte on the A100 to about 295 on the H100 and B200. A bandwidth-bound biological code gains from a new GPU roughly in proportion to bandwidth, not to the headline throughput.

4 Workloads

4.1 Classical molecular dynamics

Classical molecular dynamics integrates Newton's equations for every atom of a solvated protein, typically at femtosecond steps for millions to billions of steps. The leading GPU codes do not need double precision for this. AMBER's GPU engine computes forces in single precision and accumulates them in 64-bit fixed-point integers, which reproduces double-precision energy conservation at single-precision speed [11]. GROMACS runs most of its arithmetic in single precision and splits work between CPU and GPU [12]. OpenMM 8 adds machine-learned interatomic potentials, which bring tensor-unit work into an otherwise FP32 code [13].

A typical simulation of one protein holds tens to hundreds of thousands of atoms and needs a few gigabytes of memory. It cannot use a B200's 180 GB, and its speed is limited by FP32 throughput and by latency between steps rather than by tensor arithmetic. A systematic benchmark of

GROMACS found that consumer-class GPUs delivered the most simulated time per unit of hardware cost [14]. At the other extreme, Anton 3 shows how far hardware built for this one task can go [15]. For most labs the efficient choice is many inexpensive GPUs with strong FP32, or a large GPU partitioned into isolated slices. The B200 supports up to seven such partitions of about 23 GB each [3].

4.2 Electronic structure and other double-precision codes

Density functional theory, coupled-cluster methods and many continuum solvers rely on FP64 for numerical stability. For these codes, Figure 1 is a warning. The H100 and H200 remain NVIDIA's strongest native FP64 parts per watt.

There is a partial remedy. The Ozaki scheme splits each double-precision number into slices that low-precision integer or floating-point units can multiply exactly, then recombines the products [16]. NVIDIA's cuBLAS library has offered FP64 matrix multiplication by this route since CUDA 13.0 Update 2, with an automatic check that falls back to native FP64 when the inputs would lose accuracy [17]. A recent extension reports accuracy indistinguishable from native double-precision matrix multiplication with under 10 percent overhead, and a 2.3-fold speed-up over native cuBLAS on a GB200 in favourable cases [18]. The remedy covers dense matrix multiplication. Codes whose double-precision time is spent elsewhere, in fast Fourier transforms, sparse operations or bandwidth-bound loops, gain little. Before buying Blackwell Ultra for chemistry, a lab should profile how much of its FP64 time is spent in dense matrix products.

4.3 Protein structure prediction

AlphaFold 2 made accurate structure prediction routine [19], and AlphaFold 3 extended it to complexes with nucleic acids, ligands and modified residues [20]. Both keep a representation of every pair of tokens, so memory grows with the square of input size. AlphaFold 3's developers document that inputs of up to 5,120 tokens fit on one 80 GB A100 or H100, that a 40 GB A100 reaches 4,352 tokens only with unified memory and a changed sharding configuration, and that larger inputs must spill to host memory at a cost in speed [21]. Memory capacity, not arithmetic, sets the largest complex a lab can fold on one GPU. That makes the 141 GB H200 and 180 GB B200 attractive here for reasons that have nothing to do with tensor throughput.

The same documentation reports compile-free inference times of 2,547 seconds on an A100 and 1,416 seconds on an H100 at 5,120 tokens [21]. The H100 is 1.8 times faster. Section 5 shows that it is not much more efficient.

AlphaFold 3 is also not only a GPU job. Its genetic search stage runs on CPUs, needs at least 64 GB of host memory for long targets, and reads databases of about 630 GB that its developers recommend holding on solid-state storage [22]. A cluster tuned for language models, with expensive GPUs and thin CPU and storage, can leave those GPUs waiting.

4.4 Biological foundation models

Protein and genome language models are the one biological workload that looks like the workloads GPUs are now built for. ESM-2 and ESMFold showed that a language model trained on protein sequences can predict structure at atomic resolution [23]. ESM3, at 98 billion parameters, was trained with 1.07×10^{24} floating-point operations [24]. Evo modelled DNA at genome scale [25], and the largest Evo 2 model has 40 billion parameters and a context of up to one million nucleotides [26]. NVIDIA reports that it was trained on 2,048 H100 GPUs [27].

Training at this scale uses dense tensor arithmetic, fast links between GPUs and large memory, and it benefits fully from each new generation. Serving these models depends on capacity. NVIDIA's packaged Evo 2 40B service is tested on one H200 or on two H100s [28], because the model and its long-context state must fit in memory.

4.5 Genomics

Secondary analysis of sequencing data, meaning alignment and variant calling, is the most mature GPU workload in biology. In a cloud benchmark of NVIDIA Parabricks, GPU versions of germline callers ran up to 65 times faster than their CPU versions, bringing GATK HaplotypeCaller from 36 hours to 33 minutes on Amazon Web Services and 24 minutes on an on-premises NVIDIA DGX [29]. The same study found that some somatic callers cost more on GPUs than on CPUs, because their acceleration did not cover the higher hourly price [29]. Deep-learning variant callers such as DeepVariant add convolutional inference to the pipeline [30].

These pipelines move large files and run modest neural networks. They reward storage throughput and cost per genome, and they run well on mid-range GPUs. They need neither the largest memory nor the highest tensor throughput.

4.6 Cryo-electron microscopy

Single-particle cryo-EM reconstructs three-dimensional maps from hundreds of thousands to millions of noisy particle images. GPU acceleration in RELION-2 cut classification and refinement times enough to make high-resolution processing practical on a single workstation [31], and cryoSPARC built its algorithms around GPUs from the start [32]. The arithmetic is mostly FP32 and FFT-based. Memory needs grow with box size and the number of classes. The binding constraints are FP32 throughput, memory capacity for large boxes, and storage for particle stacks, which places cryo-EM close to molecular dynamics in its hardware needs.

4.7 Imaging and pathology foundation models

Pathology now has its own foundation models. UNI was pretrained on more than 100 million tissue patches from over 100,000 whole-slide images [33]. Prov-GigaPath was pretrained on 1.3 billion tiles from 171,189 slides, with a slide-level encoder that treats a whole gigapixel slide as one long sequence of tiles [34]. Training these models is a large-model workload with long sequences, which favours memory capacity. Applying them is dominated by encoding thousands of tiles per slide, which is high-throughput tensor work on modest GPUs, and by reading the slides, which is an input and output problem.

4.8 Optical and acoustic simulation

Biophotonics depends on two kinds of simulation that run very differently. Monte Carlo photon transport tracks millions of independent photon packets through scattering tissue, an embarrassingly parallel FP32 computation dominated by random-number generation and branching, which GPUs accelerated by orders of magnitude [35]. It uses no tensor units at all. Acoustic propagation for photoacoustic imaging is commonly computed with k-space pseudospectral methods [36], which are dominated by fast Fourier transforms over three-dimensional grids and limited by memory bandwidth and capacity. The author's earlier work on instrument transfer for Raman models [37] and on specifying photoacoustic front-end chips from clinical targets [38] rests on calculations of exactly these two kinds. Neither benefits from the precision formats that dominate recent GPU gains.

5 Energy per unit of work

US data centres used about 176 TWh of electricity in 2023, 4.4 percent of national consumption, and Lawrence Berkeley National Laboratory projects 325 to 580 TWh by 2028 [39]. Within that total, the energy of a scientific job depends on how well the job's arithmetic matches the chip.

Work	Chip	Energy	Basis
10^{18} FP64 operations at peak	H100	2.9 kWh	67 TFLOPS at 700 W
	B200	7.5 kWh	37 TFLOPS at 1,000 W
	B300 (GB300)	c. 280 kWh	1.39 TFLOPS at 1,400 W
10^{18} BF16 operations at peak	H100	0.20 kWh	989 TFLOPS at 700 W
	B200	0.12 kWh	2,250 TFLOPS at 1,000 W
AlphaFold 3, one 5,120-token input	A100	0.28 kWh	2,547 s at 400 W
	H100	0.28 kWh	1,416 s at 700 W
Training at ESM3 scale, 1.07×10^{24} operations	H100	c. 1.2 GWh	40% of peak, DGX power, PUE 1.3
	B200	c. 0.8 GWh	same assumptions

Table 2. Energy per unit of work, from Table 1, AlphaFold 3 timings [21] and ESM3's reported training compute [24]. GPU energy uses rated power, an upper bound on actual draw. The training rows use DGX system power per GPU (1.275 kW for the H100, 1.79 kW for the B200) and a facility overhead of 1.3. The 40 percent utilisation is an assumption, not a reported figure, and ESM3's actual hardware is not used here.

Table 2 shows three patterns. For low-precision tensor work, each generation cuts energy. Training a model of ESM3's size would take about 0.75 million H100-hours or 0.33 million B200-hours at the assumed utilisation, and the B200 would use about 38 percent less energy. For native double-precision work the direction reverses, and Blackwell Ultra makes it two orders of magnitude worse unless the code can use emulation. For memory-bound work the gain is small. AlphaFold 3 runs 1.8 times faster on the H100 than on the A100, but because the H100's rated power is 1.75 times higher, the GPU energy per prediction falls by only 3 percent. Using whole-server power instead of GPU power, the saving is 13 percent.

The general point is that energy efficiency is a property of the pairing between a workload and a chip, not of the chip alone. A biology department that measures its footprint per GPU-hour will miss most of the variation that matters.

6 Matching hardware to work

Table 3 collects the results. Three recommendations follow for a research organisation planning its computing.

Profile before buying. The share of runtime spent in tensor-unit arithmetic, non-tensor FP32, FP64 and memory traffic determines which generation pays. Few published biology workloads report this breakdown.

Separate the fleet. Foundation-model training is the one workload that justifies the newest, most expensive parts, and it is bursty enough to rent. Simulation, cryo-EM, genomics and inference can run continuously on Hopper-class hardware, where the price per hour is lower and which, for FP32 and FP64, is as efficient per watt as Blackwell or more so. Earlier work by the author found that older GPUs rent well above their value on arithmetic alone and remain economic for years where power is available [40]. For double-precision science, the H100 and H200 may hold

Workload	Precision	Binding resource	Best-value hardware	Watch for
Molecular dynamics	FP32 with fixed-point accumulation	FP32 throughput, step latency	Many mid-range or consumer-class GPUs, or partitioned large GPUs	Machine-learned potentials add tensor work
Electronic structure	FP64	Native FP64, or dense FP64 matrix products	H100 or H200 class; Blackwell only with emulation-ready libraries	Share of FP64 time outside dense matrix products
Structure prediction	Mixed, mostly 32 and 16 bit	GPU memory capacity; CPU and storage for search	80 GB or larger GPUs, balanced with CPU cores, host memory and fast storage	Pair memory grows with the square of input size
Foundation-model training	BF16, FP8	Tensor throughput, interconnect, memory	Newest generation, rented by the hour or the month	Cost of idle capacity between runs
Foundation-model inference	BF16, FP8	Memory capacity and bandwidth	Large-memory GPUs such as H200 or B200	Long-context state size
Genomics	Mixed	Storage throughput, cost per genome	Mid-range GPUs	Some callers cost more on GPU than CPU
Cryo-EM	FP32	FP32 throughput, memory for large boxes, storage	Mid-range to large-memory GPUs	Particle-stack storage
Pathology models	BF16	Memory for training; input and output for inference	Large GPUs to train, modest GPUs to apply	Gigapixel slide reading
Photon and acoustic simulation	FP32	FP32 throughput (Monte Carlo); bandwidth and memory (FFT)	Mid-range GPUs with high FP32 and bandwidth per dollar	Grid size in three dimensions

Table 3. A matching table drawn from Sections 3 to 5.

their value longest.

Account for energy per job. Reporting kilowatt-hours per simulated microsecond, per predicted structure or per genome would let funders and institutions compare groups and hardware on the measure that matters.

7 Limits

Every throughput figure here is a vendor peak. Real codes reach a fraction of peak that varies by workload and by generation, and nothing in this paper measures it directly. Energy figures use rated power, which overstates actual draw for memory-bound jobs. The workload descriptions summarise published requirements and benchmarks from different years, hardware and software versions. The ESM3-scale training estimate rests on an assumed utilisation. AMD, Google and other accelerators are not covered, and some of them keep stronger double-precision units. A direct benchmark of representative biological codes across the six GPUs, with measured power, would test the conclusions here and is the obvious next step.

Funding and interests

No external funding was received. The author has no financial interest in any company named.

References

[1] NVIDIA Corporation (2025). Inside NVIDIA Blackwell Ultra: The chip powering the AI factory era. NVIDIA Technical Blog, August 2025. https://developer.nvidia.com/blog/inside-nvidia-blackwell-ultra-the-chip-powering-the-ai-factory-era/

[2] Tom's Hardware (2025). Nvidia shares Blackwell Ultra's secrets: NVFP4 boost detailed and PCIe 6.0 support. https://www.tomshardware.com/pc-components/gpus/nvidia-shares-blackwell-ultras-secrets-nvfp4-boost-detailed-and-pcie-6-0-support

[3] Lenovo Press (2025). ThinkSystem NVIDIA HGX B200 180GB 1000W GPU, product guide LP2226. https://lenovopress.lenovo.com/lp2226-thinksystem-nvidia-b200-180gb-1000w-gpu

[4] Williams, S., Waterman, A., and Patterson, D. (2009). Roofline: An insightful visual performance model for multicore architectures. *Communications of the ACM*, 52(4), 65-76.

[5] Gholami, A., Yao, Z., Kim, S., Hooper, C., Mahoney, M. W., and Keutzer, K. (2024). AI and memory wall. *IEEE Micro*, 44(3), 33-39.

[6] NVIDIA Corporation (2017). NVIDIA Tesla V100 GPU datasheet. https://www.nvidia.com/en-us/data-center/v100/

[7] NVIDIA Corporation (2021). NVIDIA A100 Tensor Core GPU datasheet. https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/a100/pdf/nvidia-a100-datasheet-nvidia-us-2188504-web.pdf

[8] NVIDIA Corporation (2023). NVIDIA H100 Tensor Core GPU datasheet. https://www.nvidia.com/en-us/data-center/h100/

[9] NVIDIA Corporation (2024). NVIDIA H200 Tensor Core GPU. https://www.nvidia.com/en-us/data-center/h200/

[10] NVIDIA Corporation (2025). NVIDIA GB300 NVL72. https://www.nvidia.com/en-us/data-center/gb300-nvl72/

[11] Le Grand, S., Götz, A. W., and Walker, R. C. (2013). SPFP: Speed without compromise. A mixed precision model for GPU accelerated molecular dynamics simulations. *Computer Physics Communications*, 184(2), 374-380.

[12] Páll, S., Zhmurov, A., Bauer, P., Abraham, M., Lundborg, M., Gray, A., Hess, B., and Lindahl, E.

(2020). Heterogeneous parallelization and acceleration of molecular dynamics simulations in GROMACS. *Journal of Chemical Physics*, 153(13), 134110.

[13] Eastman, P., Galvelis, R., Peláez, R. P., et al. (2024). OpenMM 8: Molecular dynamics simulation with machine learning potentials. *Journal of Physical Chemistry B*, 128(1), 109-116.

[14] Kutzner, C., Páll, S., Fechner, M., Esztermann, A., de Groot, B. L., and Grubmüller, H. (2019). More bang for your buck: Improved use of GPU nodes for GROMACS 2018. *Journal of Computational Chemistry*, 40(27), 2418-2431.

[15] Shaw, D. E., Adams, P. J., Azaria, A., et al. (2021). Anton 3: Twenty microseconds of molecular dynamics simulation before lunch. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC21)*. https://doi.org/10.1145/3458817.3487397

[16] Ozaki, K., Ogita, T., Oishi, S., and Rump, S. M. (2012). Error-free transformations of matrix multiplication by using fast routines of matrix multiplication and its applicability to dot products. *Numerical Algorithms*, 59(1), 95-118.

[17] NVIDIA Corporation (2025). Unlocking tensor core performance with floating point emulation in cuBLAS. NVIDIA Technical Blog. https://developer.nvidia.com/blog/unlocking-tensor-core-performance-with-floating-point-emulation-in-cublas/

[18] Schwarz, A., Anders, A., Brower, C., Bayraktar, H., et al. (2026). Guaranteed DGEMM accuracy while using reduced precision tensor cores through extensions of the Ozaki scheme. In *SCA/HPCAsia 2026*. https://doi.org/10.1145/3773656.3773670

[19] Jumper, J., Evans, R., Pritzel, A., et al. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, 596, 583-589.

[20] Abramson, J., Adler, J., Dunger, J., et al. (2024). Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, 630, 493-500.

[21] Google DeepMind (2024). AlphaFold 3: Performance and installation documentation. https://github.com/google-deepmind/alphafold3/blob/main/docs/performance.md

[22] Google DeepMind (2024). AlphaFold 3: Installation and running your first prediction. https://github.com/google-deepmind/alphafold3/blob/main/docs/installation.md

[23] Lin, Z., Akin, H., Rao, R., et al. (2023). Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637), 1123-1130.

[24] Hayes, T., Rao, R., Akin, H., et al. (2025). Simulating 500 million years of evolution with a language model. *Science*, 387(6736), 850-858.

[25] Nguyen, E., Poli, M., Durrant, M. G., et al. (2024). Sequence modeling and design from molecular to genome scale with Evo. *Science*, 386(6723), eado9336.

[26] Brixi, G., Durrant, M. G., Ku, J., et al. (2025). Genome modeling and design across all domains of life with Evo 2. bioRxiv 2025.02.18.638918.

[27] NVIDIA Corporation (2025). Understanding the language of life's biomolecules across evolution at a new scale with Evo 2. NVIDIA Technical Blog, 19 February 2025. https://developer.nvidia.com/blog/understanding-the-language-of-lifes-biomolecules-across-evolution-at-a-new-scale-with-evo-2/

[28] NVIDIA Corporation (2025). Evo 2 40B model card. https://build.nvidia.com/arc/evo2-40b/modelcard

[29] O'Connell, K. A., Yosufzai, Z. B., Campbell, R. A., et al. (2023). Accelerating genomic workflows using NVIDIA Parabricks. *BMC Bioinformatics*, 24, 221.

[30] Poplin, R., Chang, P.-C., Alexander, D., et al. (2018). A universal SNP and small-indel variant caller using deep neural networks. *Nature Biotechnology*, 36, 983-987.

[31] Kimanius, D., Forsberg, B. O., Scheres, S. H. W., and Lindahl, E. (2016). Accelerated cryo-EM structure determination with parallelisation using GPUs in RELION-2. *eLife*, 5, e18722.

[32] Punjani, A., Rubinstein, J. L., Fleet, D. J., and Brubaker, M. A. (2017). cryoSPARC: Algorithms for rapid unsupervised cryo-EM structure determination. *Nature Methods*, 14, 290-296.

[33] Chen, R. J., Ding, T., Lu, M. Y., et al. (2024). Towards a general-purpose foundation model for computational pathology. *Nature Medicine*, 30, 850-862.

[34] Xu, H., Usuyama, N., Bagga, J., et al. (2024). A whole-slide foundation model for digital pathology from real-world data. *Nature*, 630, 181-188.

[35] Fang, Q., and Boas, D. A. (2009). Monte Carlo simulation of photon migration in 3D turbid media accelerated by graphics processing units. *Optics Express*, 17(22), 20178-20190.

[36] Treeby, B. E., and Cox, B. T. (2010). k-Wave: MATLAB toolbox for the simulation and reconstruction of photoacoustic wave fields. *Journal of Biomedical Optics*, 15(2), 021314.

[37] Sadineni, A. C. (2026). Spectral Passport: When does a Raman model's uncertainty guarantee survive a change of instrument? Preprint, November 2026. https://doi.org/10.5281/zenodo.24210927

[38] Sadineni, A. C. (2026). Outcome-defined silicon: Specifying front-end chips for cancer imaging from the clinical decision backward. Preprint, November 2026. https://doi.org/10.5281/zenodo.24210908

[39] Shehabi, A., Smith, S. J., Hubbard, A., Newkirk, A., et al. (2024). 2024 United States Data Center Energy Usage Report. Lawrence Berkeley National Laboratory, LBNL-2001637.

[40] Sadineni, A. C. (2026). The economic life of an accelerator: Rental prices, power and the six-year depreciation of data-centre GPUs. Preprint, November 2026. https://doi.org/10.5281/zenodo.24210935