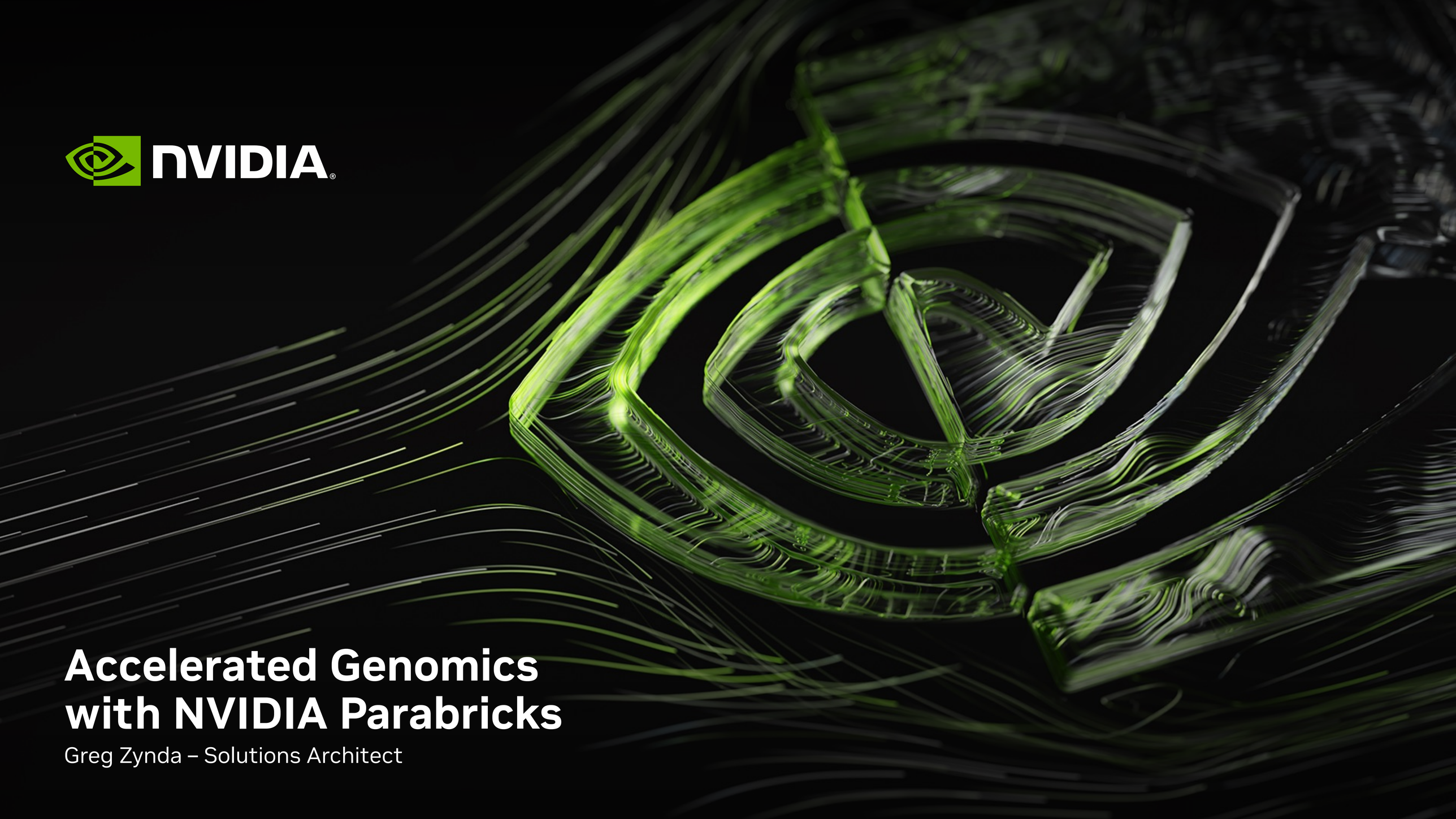


# Accelerated Genomics with NVIDIA Parabricks

Greg Zynda – Solutions Architect

# NVIDIA Clara AI Computing Platform

## $10T Global Healthcare Expenditure | World's Largest Data Industry by 2025

IMAGING | MEDICAL DEVICE | GENOMICS | DRUG DISCOVERY

NVIDIA CLARA


PLATFORMS

NVIDIA AI

NVIDIA Omniverse

CHIPS & SYSTEMS

GPU | DPU | CPU | IGX | EGX | DGX

Source: Global spending on health: rising to the pandemic's challenges. Geneva: World Health Organization; 2022. Licence: CC BY-NC-SA 3.0 IGO

IDC White Paper, Doc# US44413318, November 2018: The. Digitization of the World – From Edge to Core"

NVIDIA

# NVIDIA ACROSS THE FULL GENOMICS WORKFLOW

All the tools required for full stack analysis


E2E Genomic Analysis Workflow

## PRIMARY ANALYSIS

Classifying raw signal from the sequencer into As, Cs, Gs, & Ts (often Deep Learning based)

## SECONDARY ANALYSIS

Assembly/alignment of the ACGT reads from the sequencer, and calling variants

## TERTIARY ANALYSIS

Multi-sample investigations and/or interpretation of results

TensorRT

TensorFlow

PyTorch

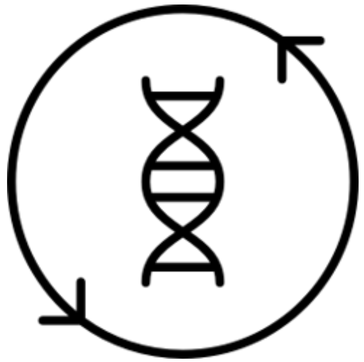
NVIDIA CLARA PARABRICKS

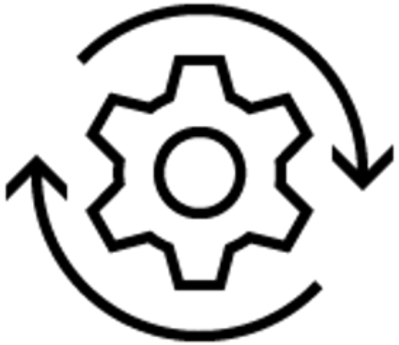
RAPIDS

NVIDIA AI Enterprise Supported

# NVIDIA Parabricks

## Secondary Analysis with Speed, Accuracy, Flexibility





### Key Applications

Accelerating tooling for gold-standard germline, somatic and RNA analysis, at speed.

### Up to 80x Acceleration

Up to 80x faster for WGS than CPU-only solutions, reducing computing costs by up to 50 percent.

### Flexible Workflows

Create powerful customized workflows by configuring tools in WDL and NextFlow.

### Better Accuracy

Bring the power of deep learning to your genomic analysis with Clara Parabricks and GPUs.

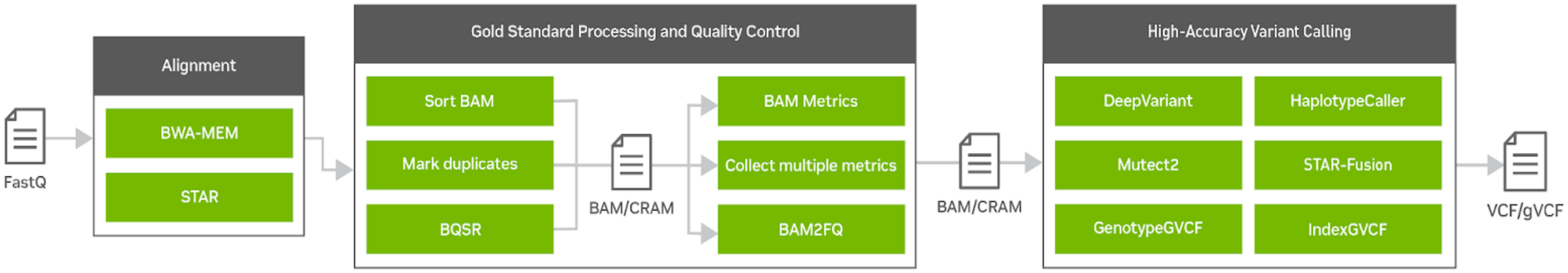
# Key Applications of Clara Parabricks

## Accelerated and Deep Learning Genomic Analysis

**Population Genomics**

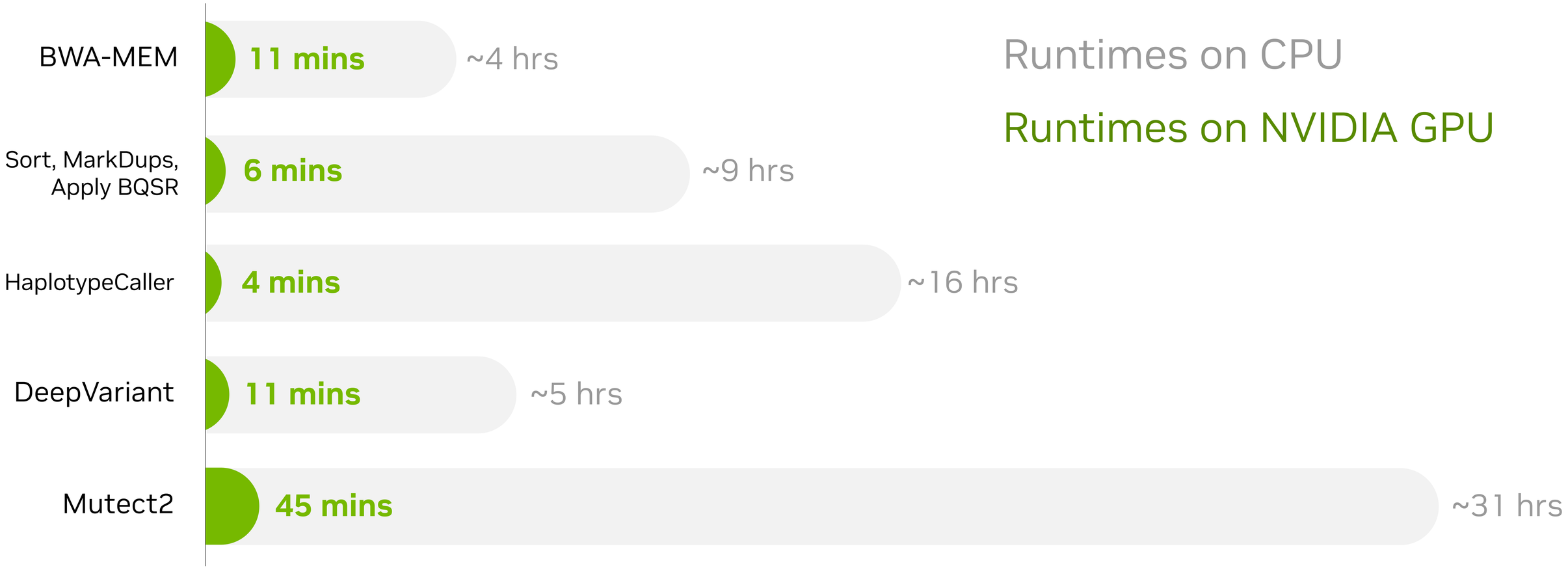
**Cancer Genomics**

**RNA Sequencing**



# Up to 80x Acceleration

Gold-standard results, faster

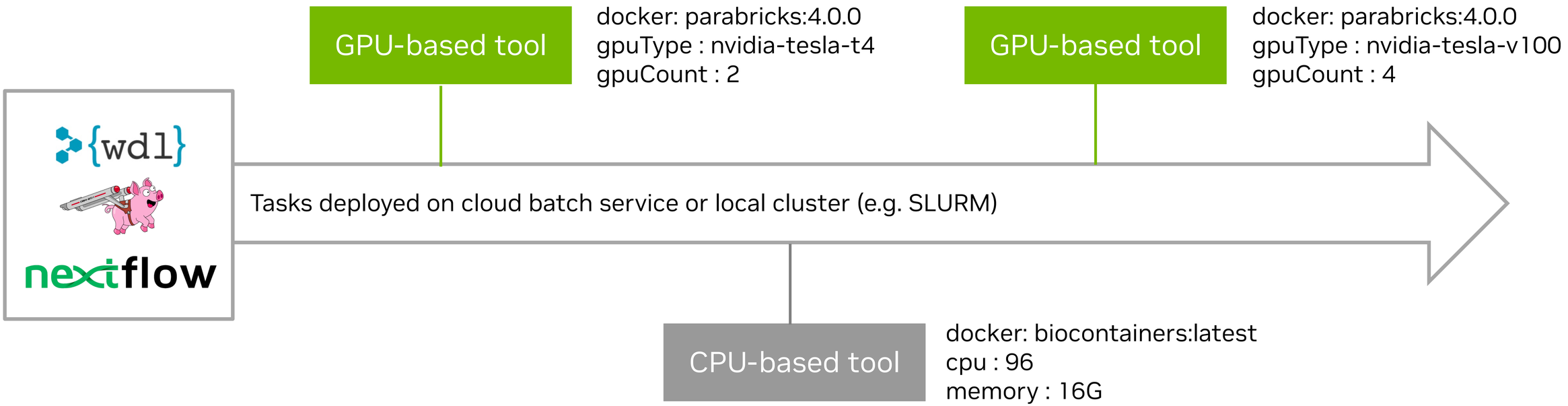



**v3.8 Benchmarks**
Dataset: HG002 30x WGS, except Mutect2 on SEQC2 50x WGS
CPU: m5.24xlarge; GPU: 8xA100, except DeepVariant & Mutect2 on 8xV100

NVIDIA

# Workflow Manager Compatible

## Customize and deploy Parabricks at scale




Clara Parabricks is fully compatible with common workflow managers WDL and NextFlow for deploying at scale

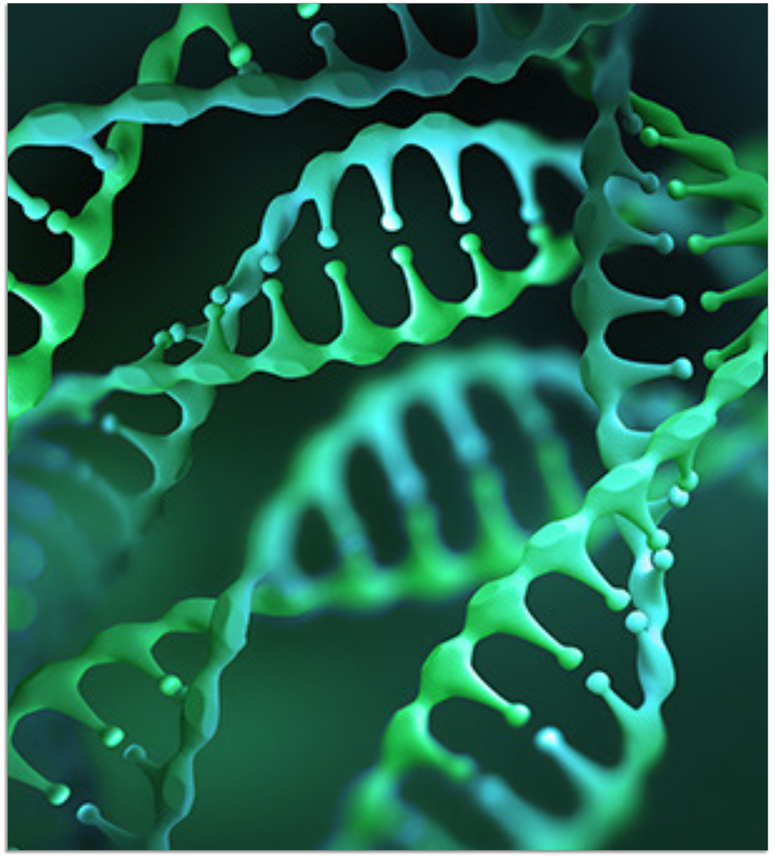
- Intertwine GPU and CPU powered tasks with different compute requirements
- Reference workflows and recommended compute configs at: github.com/clara-parabricks-workflows

# Free to Use, plus NVIDIA AI Support for Enterprise

| | Free | NVIDIA AI Enterprise |
|---|---|---|
| **Accelerated tools and high throughput workflows** | ✓ | ✓ |
| **Community forum support**<br>Submit questions to the Clara Parabricks Developers Forum for community support | ✓ | ✓ |
| **Full stack & business-critical support**<br>Enterprise-grade support for multiple deployment options: bare-metal, containerized, cloud, and more | | ✓ |
| **Access to NVIDIA experts**<br>Guidance on configuration and performance, including access to NVIDIA engineers | | ✓ |
| **Enterprise training services**<br>Instructor-led workshops and self-paced training for developers, data scientists, and IT professionals | | ✓ |
| **Security notifications**<br>Receive priority notifications of the latest security fixes and maintenance releases. | | ✓ |
| **Support for other NVIDIA software and SDKs**<br>NVIDIA AI Enterprise covers support for many NVIDIA products, including RAPIDS, TensorRT, and more | | ✓ |

# Making a Real-World Impact

## Clara Parabricks for sequencing centers, national programs, and groundbreaking research

### Human Genome Center of Tokyo

Deploying rapid analysis of human genomes with Clara Parabricks and 80 NVIDIA V100 GPUs in their SHIROKANE compute cluster

### Regeneron & UKBioBank

Regeneron Genetics Center has sequenced the exomes of all 450,000 participants of UKBIoBank, processing the vast data with Clara Parabricks

### Stanford University World Record

World record for fastest DNA sequencing, set in collaboration with NVIDIA, using Clara Parabricks' accelerated DeepVariant

# Getting Started with Clara Parabricks

# Running Clara Parabricks

## Requirements

### Hardware Requirements

- Any NVIDIA GPU that supports CUDA architecture 60, 70, 75, or 80 and has at least 16GB of GPU RAM. Parabricks has been tested on the following NVIDIA GPUs:
  - V100
  - T4
  - A10, A30, A40, A100, A6000
- System Requirements:
  - A 2 GPU server should have at least 100GB CPU RAM and at least 24 CPU threads.
  - A 4 GPU server should have at least 196GB CPU RAM and at least 32 CPU threads.
  - A 8 GPU server should have at least 392GB CPU RAM and at least 48 CPU threads.

**Note**

Clara Parabricks is not supported on virtual (vGPU) or Multi-Instance (MIG) GPUs.

**Note**

The Clara Parabricks deepvariant and deepvariant_germline tools ship with support for T4, V100, and A100 GPUs. See the Models for additional GPUs section for more details on downloading model files for A10, A30, A40, A100, and A6000 GPUs for the deepvariant and deepvariant_germline tools.

### Software Requirements

The following are software requirements for running Clara Parabricks.

- An NVIDIA driver greater than version 465.32.* .
- Any Linux Operating System that supports nvidia-docker2 Docker version 20.10 (or higher)

### Verifying Hardware and Software Requirements

#### Checking available NVIDIA hardware and driver

To check your NVIDIA hardware and driver version, use the `nvidia-smi` command:

```
$ nvidia-smi
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 515.65.01    Driver Version: 515.65.01    CUDA Version: 11.7     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla V100-DGXS...  On   | 00000000:07:00.0 Off |                    0 |
| N/A   44C    P0    38W / 300W |     74MiB / 16155MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|    0   N/A  N/A      3019      G   /usr/lib/xorg/Xorg                 56MiB |
+-----------------------------------------------------------------------------+
```

This shows the following important information:

- The NVIDIA driver version is 515.65.01.
- The supported CUDA driver API is 11.7.
- The GPU has 16 GB of memory.

#### Checking available CPU RAM and threads

To see how much RAM and CPU threads in your machine, you can run the following:

```
# To check available memory
$ cat /proc/meminfo | grep MemTotal

# To check available number of threads
$ cat /proc/cpuinfo | grep processor | wc -l
```

# Running Clara Parabricks

## Where


On-Premise


Public Cloud Providers

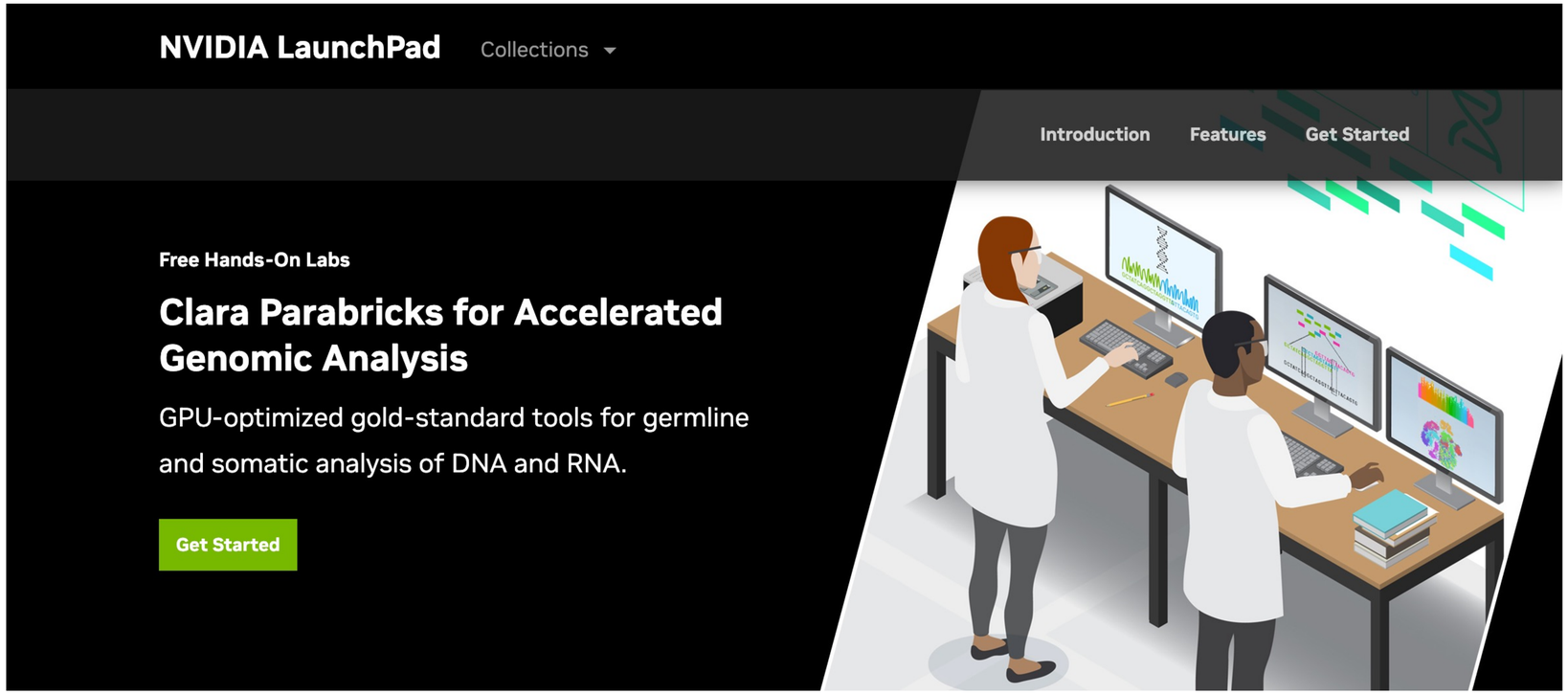

Platforms & Partners

NVIDIA NGC Catalog

All Clara Parabricks Containers are available publicly in the NGC Catalog

# Running Clara Parabricks

## Launchpad




### In this free hands-on lab, you'll experience:

Running the accelerated gold-standard tools of NVIDIA Clara™ Parabricks®, including BWA-MEM and Haplotype Caller, in a GPU environment

Powerful analysis with up to 80X acceleration over CPU-only deployment and reduced costs of up to 50 percent

How deep learning tools for genomics are improving variant calling accuracy with custom models in DeepVariant

### Ready-to-Use Infrastructure

Test and prototype on ready-to-use infrastructure that's hosted at Equinix and available to you for up to two weeks.

### A Hands-On Experience

Take curated labs that walk you through the entire process, from infrastructure optimization to application deployment.

NVIDIA

# Running Clara Parabricks

## Drop-in Command Line Replacements

### Getting the Software

The Clara Parabricks Docker image can be obtained by running the following command:

```
$ docker pull nvcr.io/nvidia/clara/clara-parabricks:4.0.0-1
```

At this point the software is ready to use.

### Running Clara Parabricks

#### From the Command Line

Clara Parabricks is deployed using a docker image. There are two parts to customizing a Parabricks run:

- Customizing Docker container specific options: These are the options that are passed to the `docker` command before the name of the container. For example, the user should mount their data directories within the Docker container by passing the `-v` option to Docker. See the Tutorials for more detailed examples.
- Parabricks specific options: These options are passed to the Parabricks command line to customize the Parabricks run. For example, you can choose which tool to run and pass tool-specific options.

For example, to run the Clara Parabricks fq2bam tool using the Docker container, use the following command:

```
$ docker run \
    --gpus all \
    --rm \
    --volume /host/data:/input_data \
    --volume /host/results:/outputdir \
    --workdir /image/input_data \
    nvcr.io/nvidia/clara/clara-parabricks:4.0.0-1 \
    pbrun fq2bam \
    --ref /input_data/Homo_sapiens_assembly38.fasta \
    --in-fq /input_data/fastq1.gz /input_data/fastq2.gz \
    --out-bam /image/outputdir/fq2bam_output.bam
```

### Quick Start

```
 # This command assumes all the inputs are in <INPUT_DIR> and all the outputs go to <OUTPU
$ docker run --rm --gpus all --volume <INPUT_DIR>:/workdir --volume <OUTPUT_DIR>:/outputdi
    -w /workdir \
    nvcr.io/nvidia/clara/clara-parabricks:4.0.0-1 \
    pbrun mutectcaller \
    --ref /workdir/${REFERENCE_FILE} \
    --tumor-name tumor \
    --in-tumor-bam /workdir/${INPUT_TUMOR_BAM} \
    --in-normal-bam /workdir/${INPUT_NORMAL_BAM} \
    --normal-name normal \
    --out-vcf /outputdir/${OUTPUT_VCF}
```

### Compatible GATK4 Command

The command below is the GATK4 counterpart of the Parabricks command above. The output from this command will be identical to the output from the above command. See the Output Comparison page for comparing the results.

```
$ gatk Mutect2 \
    -R <INPUT_DIR>/${REFERENCE_FILE} \
    --input <INPUT_DIR>/${INPUT_TUMOR_BAM} \
    --tumor-sample tumor \
    --input <INPUT_DIR>/${INPUT_NORMAL_BAM} \
    --normal-sample normal \
    --output <OUTPUT_DIR>/${OUTPUT_VCF}
```

# Running Clara Parabricks

## Tutorials

Docs » Tutorials

### Tutorials

The tutorials walk you through a simple use case for Clara Parabricks, giving a brief introduction of how it worl
You will start by downloading some sample data:

- A reference file ( `Homo_sapiens_assembly38.fasta` ) and its index
- A 'known indels' file and its index
- Two FASTQ files
- Associated index files

The tutorials then walk through the following steps:

- Alignment (FASTA + FASTQ ==> BAM)
- Variant calling (BAM ==> VCF)

The tutorials are meant to be simple and straightforward and to only cover a single, specific use case. You shou
be able to copy and paste the commands into a terminal window and get the same results as shown. The How
Tos cover more general problem solving using Clara Parabricks.

#### Steps in the Tutorial




Previous

Next


Docs » Tutorials » FQ2BAM Tutorial

### FQ2BAM Tutorial

This tutorial will show you how to run our core alignment tool, FQ2BAM, which allows you to align a FASTQ file according to GATK best practices at blazing speeds. This includes the gold-standard alignment tool BWA-MEM with inbuilt co-ordinate sorting of the output file, and optionally application of base-quality-score-recalibration and marking of duplicate reads.
The `fq2bam` tool aligns, sorts (by coordinate), and marks duplicates in paired-end FASTQ file data. The data files used in this example are taken from the sample data downloaded in the previous section.
If you execute the following command using the Clara Parabricks sample data, you should get the same results as shown here.
Before executing this command, make sure your current directory is where you extracted the sample data; it should have a **parabricks_sample** sub-directory.

```
$ docker run \
      --gpus all \
      --rm \
      --volume $(pwd):/workdir \
      --volume $(pwd):/outputdir \
    nvcr.io/nvidia/clara/clara-parabricks:v4.0.1-1 \
    pbrun fq2bam \
      --ref /workdir/parabricks_sample/Ref/Homo_sapiens_assembly38.fasta \
      --in-fq /workdir/parabricks_sample/Data/sample_1.fq.gz /workdir/parabricks_sample/Data/sample_2.fq.gz
      --out-bam /outputdir/fq2bam_output.bam

[Parabricks Options Mesg]: Checking argument compatibility
[Parabricks Options Mesg]: Automatically generating ID prefix
[Parabricks Options Mesg]: Read group created for /workdir/parabricks_sample/Data/sample_1.fq.gz and
/workdir/parabricks_sample/Data/sample_2.fq.gz
[Parabricks Options Mesg]: @RG\tID:HK3TJBCX2.1\tLB:lib1\tPL:bar\tSM:sample\tPU:HK3TJBCX2.1
[PB Info 2022-Sep-02 19:49:27] ------------------------------------------------------------------------------
[PB Info 2022-Sep-02 19:49:27] ||                 Parabricks accelerated Genomics Pipeline                 |
[PB Info 2022-Sep-02 19:49:27] ||                           Version 4.0.0-1                                |
[PB Info 2022-Sep-02 19:49:27] ||                       GPU-BWA mem, Sorting Phase-I                       |
[PB Info 2022-Sep-02 19:49:27] ------------------------------------------------------------------------------
```

# Running Clara Parabricks

## Common Questions/Issues

**Can I run Parabricks with Singularity?**

We do not officially support singularity, but we have had users successfully run the Parabricks container with Singularity. Here is a guide: https://docs.sylabs.io/guides/3.10/user-guide/singularity_and_docker.html

**Can I run multiple samples on a single node? Or with MIG?**

You can run multiple on a single node, but each job still needs to meet the minimum hardware requirements.
E.g. >2 GPUs per Parabricks job. Parabricks does not support MIG.

**Can I run Parabricks on multi-node?**

This is not currently supported. The easiest way to run at multi-node scale is to run multiple Parabricks jobs at once.

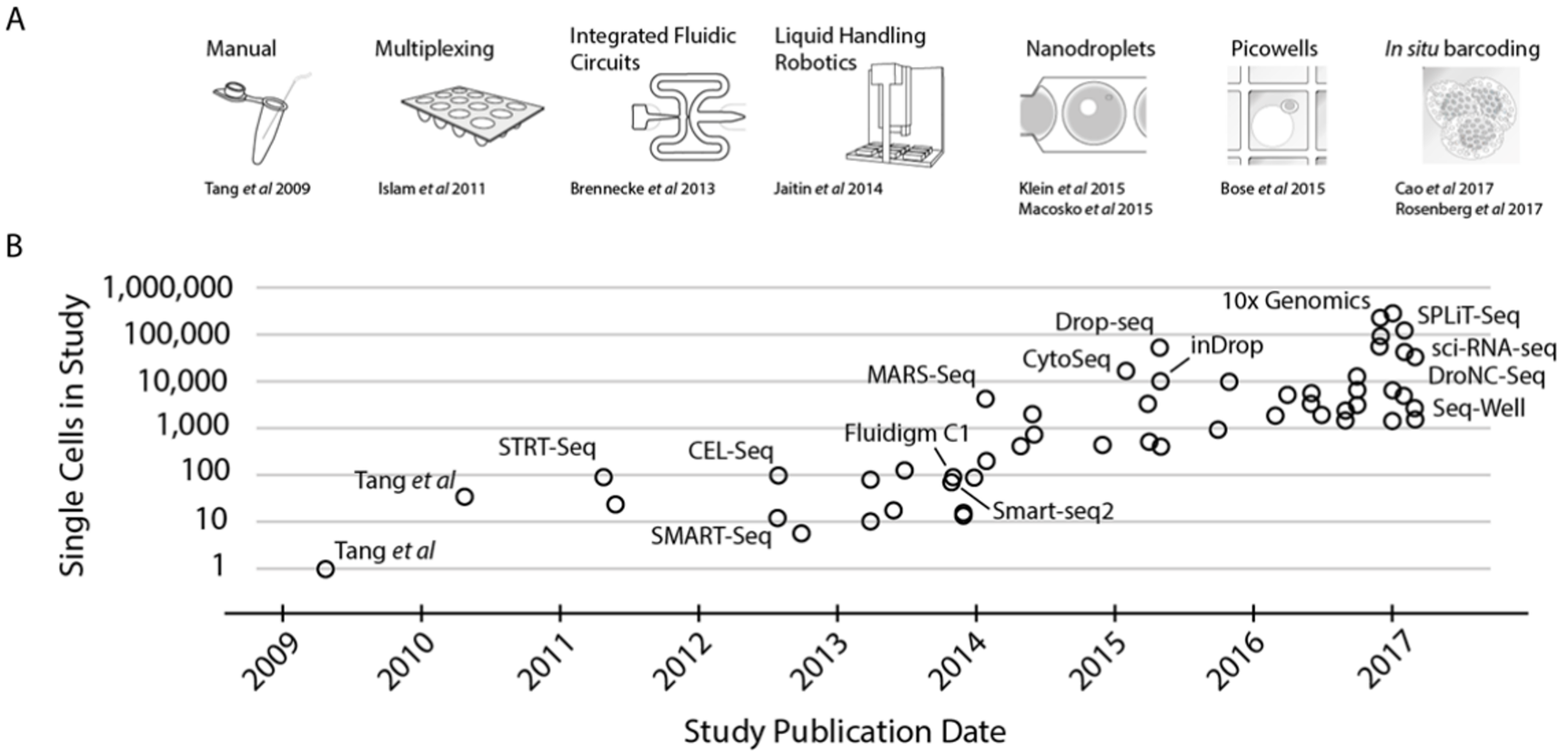
**I have the 16GB of GPU RAM required, but Parabricks says I don't have enough**

You need to have 16GB of GPU RAM *free*. Check for other jobs running on the GPU.

# Accelerating Single-Cell Genomic Analysis with RAPIDS

# A Growing Problem

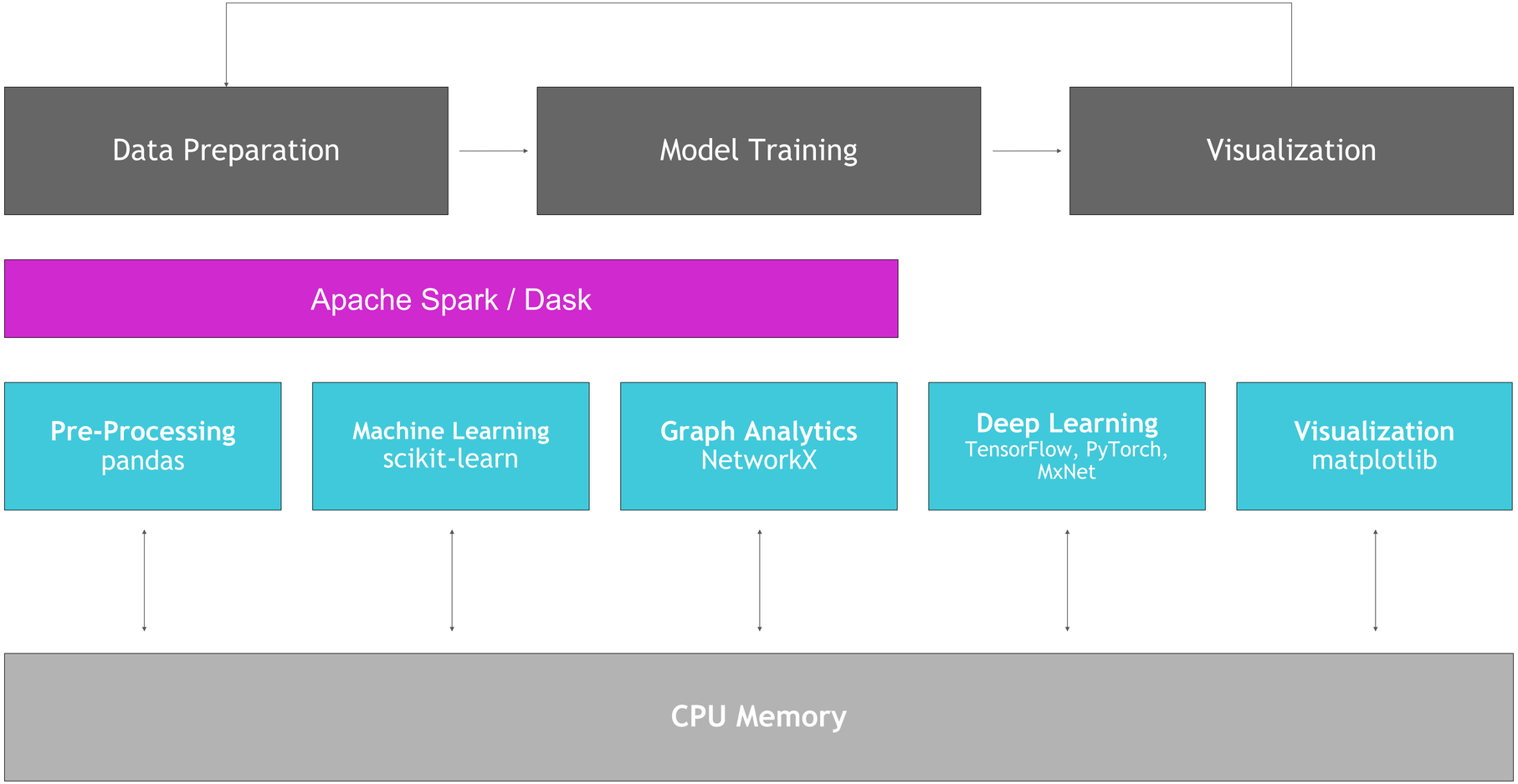
## Dataset Sizes are Growing Exponentially




Svensson V, Vento-Tormo R, Teichmann SA. Exponential scaling of single-cell RNA-seq in the past decade. Nat Protoc. 2018 Apr;13(4):599-604. doi: 10.1038/nprot.2017.149. Epub 2018 Mar 1. PMID: 29494575.

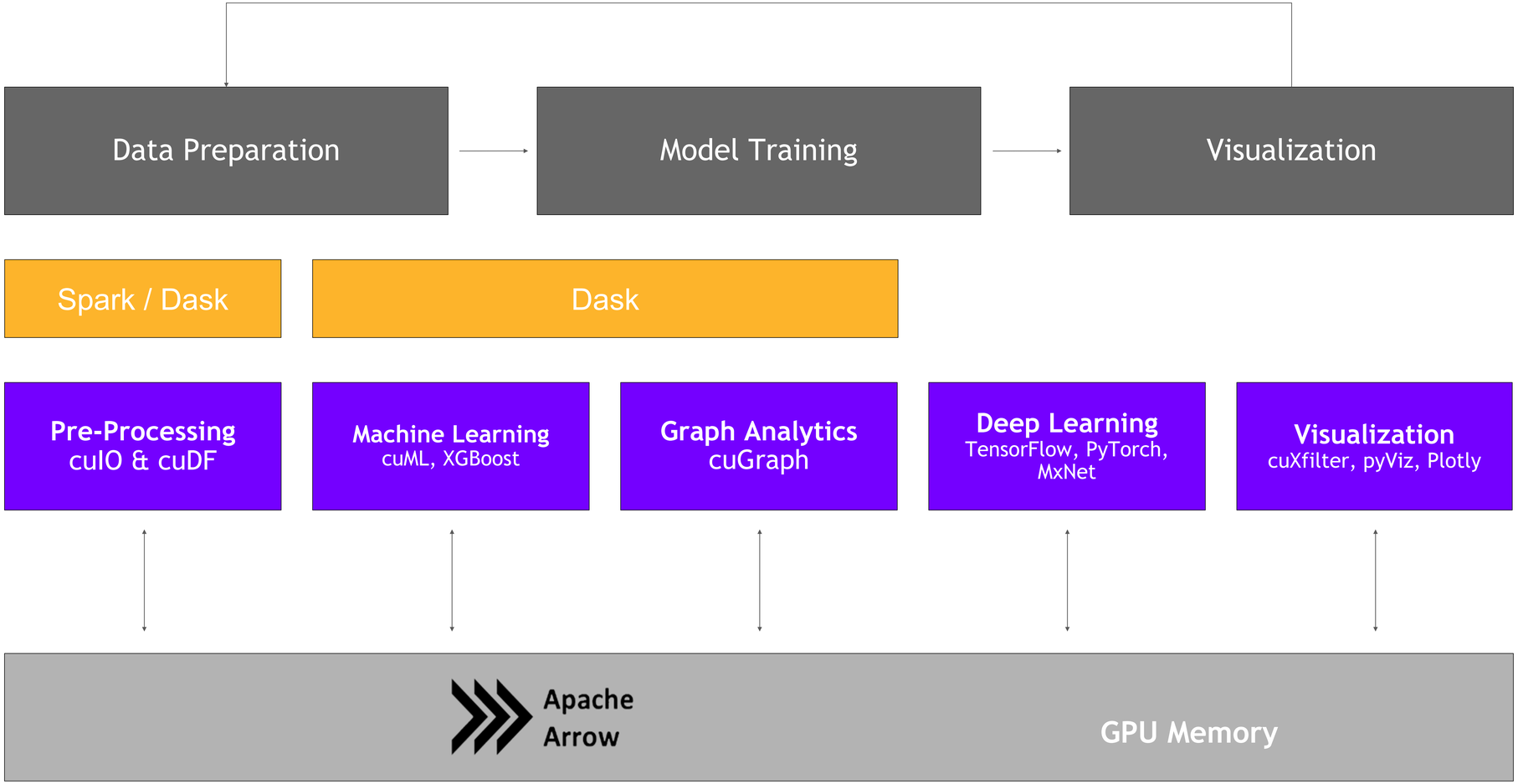
# Open Source Software Has Democratized Data Science

## Highly Accessible, Easy to Use Tools Abstract Complexity



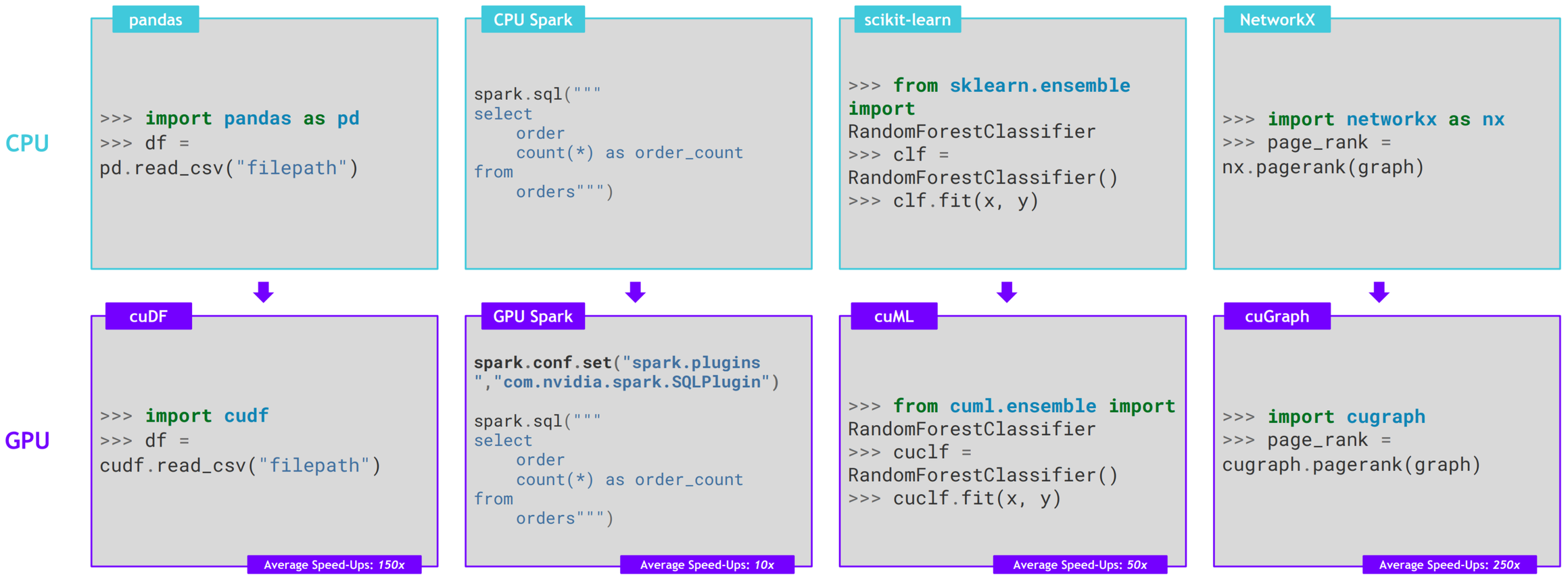
# Accelerated Data Science with RAPIDS

## Powering Popular Data Science Ecosystems with NVIDIA GPUs



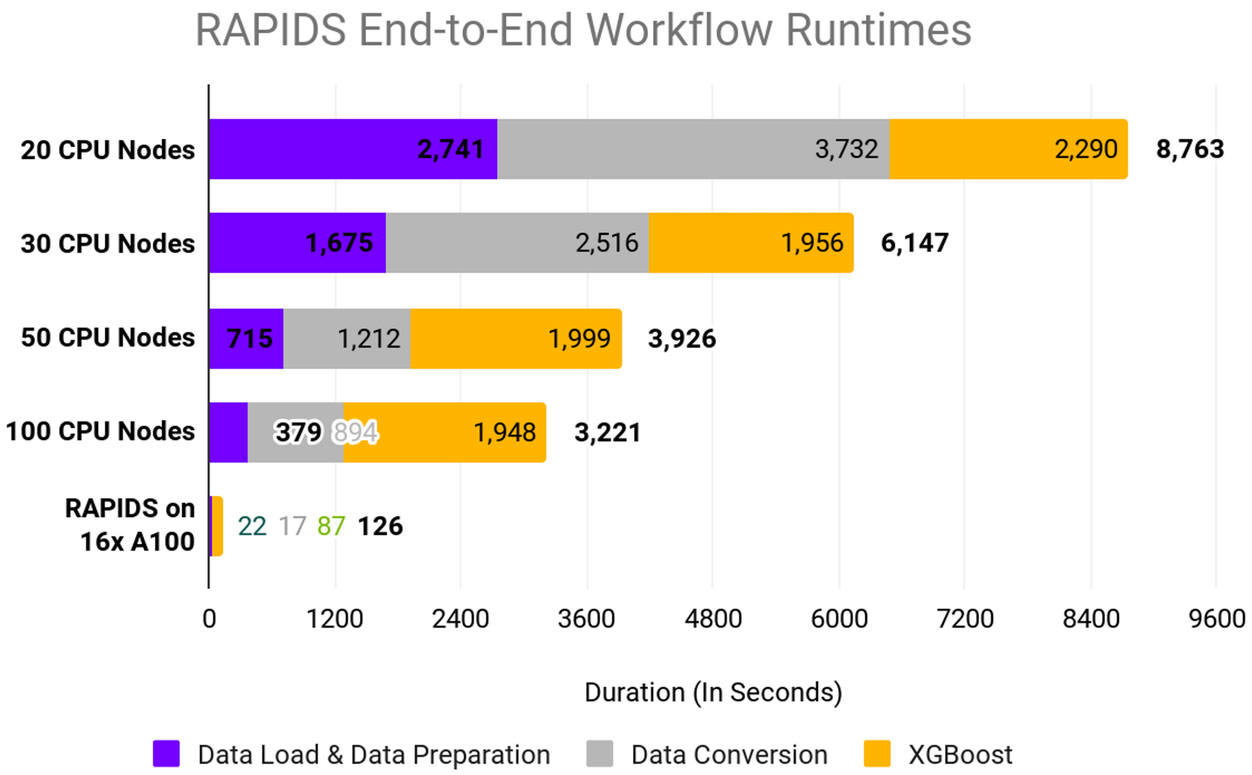
# MINOR CODE CHANGES FOR MAJOR BENEFITS

## Abstracting Accelerated Compute through Familiar Interfaces



# Lightning-Fast End-to-End Performance

## Reducing Data Science Processes from Hours to Seconds




**16** A100s Provide More Power than 100 CPU Nodes

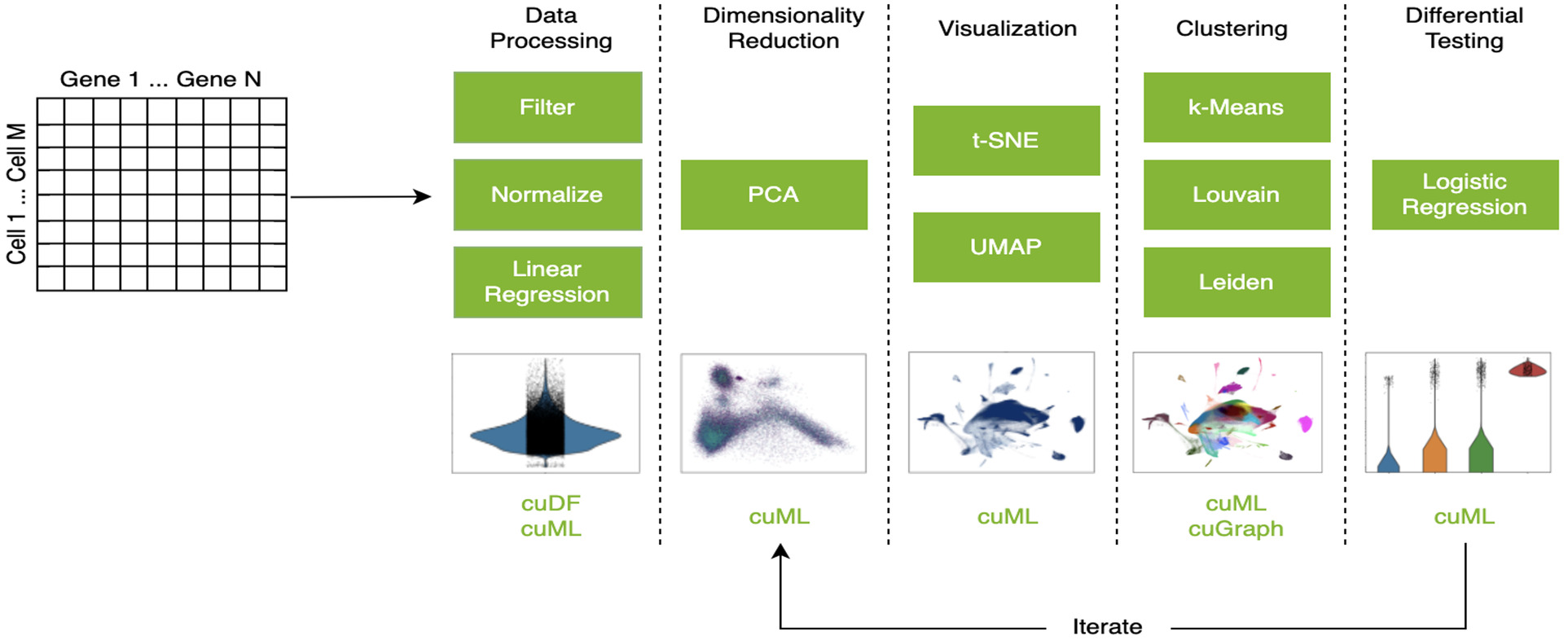
**70x** Faster Performance than Similar CPU Configuration

**20x** More Cost-Effective than Similar CPU Configuration

*CPU approximate to n1-highmem-8 (8 vCPUs, 52GB memory) on Google Cloud Platform. TCO calculations-based on Cloud instance costs.

# End-to-End Single Cell Genomics Analysis

1.3M Cells | 5 Hrs to 11 Min | $20 to $0.70



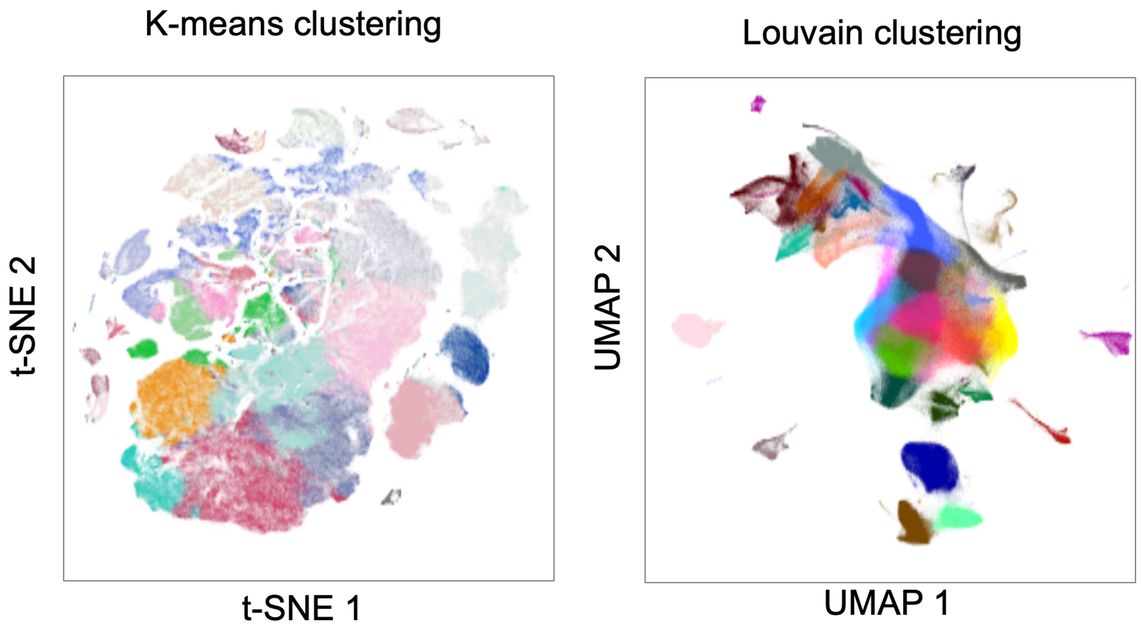

RAPIDS

scanpy

| | CPU (64 vCPUs) | GPU (T4 16GB) | GPU (A100 40GB) |
|---|---|---|---|
| Price ($/hr) | 3.786 | 1.296 | 3.673 |
| End-to-end Runtime | 5 Hrs | 30 Min (10x) | 11 Min (27x) |
| Total cost ($) | $20 | $0.63 | $0.70 |

# Analyze 1.3 million cells in 11 minutes

## Accelerating Single-Cell Analysis on a Single GPU



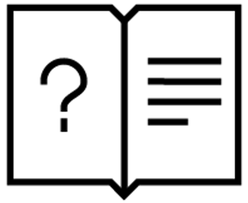

| Step | CPU n1-highmem-64 64 vCPUs | GPU n1-highmem-16 T4 16GB GPU (Acceleration) | GPU a2-highgpu-1g A100 40GB GPU (Acceleration) |
|---|---|---|---|
| Data load + Preprocessing | 1120 | 1125 (1x) | 475 (2.4x) |
| PCA | 44 | 45 (1x) | 17.8 (2.5x) |
| t-SNE | 6509 | 196 (33x) | 37 (176x) |
| k-means (single iteration) | 148 | 12.7 (12x) | 2 (74x) |
| KNN | 154 | 141 (1.1x) | 62 (2.5x) |
| UMAP | 2571 | 146 (18x) | 21 (122x) |
| Louvain clustering | 1153 | 6.1 (189x) | 2.4 (480x) |
| Leiden clustering | 6345 | 5.1 (1244x) | 1.7 (3732x) |
| Re-analysis of subgroup | 255 | 19.2 (13x) | 17.9 (14.2x) |
| End-to-end notebook run | 18338 | 1759 (10x) | 686 (27x) |
| Price ($/hr) | 3.786 | 1.296 | 3.673 |
| Total cost ($) | 19.285 | 0.633 | 0.700 |

**GPU-Accelerated Single-Cell Genomics Analysis with RAPIDS Git-Hub repo**

- 1M_brain_gpu_analysis_uvm.ipynb

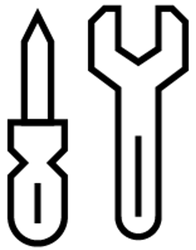
# How to Get Started with RAPIDS

## A Variety of Ways to Get Up & Running



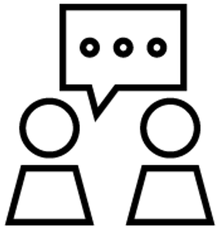
### More about RAPIDS

- Learn more at RAPIDS.ai
- Read the API docs
- Check out the RAPIDS blog
- Read the NVIDIA DevBlog



### Self-Start Resources

- Get started with RAPIDS
- Deploy on the Cloud today
- Start with Google Colab
- Look at the cheat sheets



### Discussion & Support

- Check the RAPIDS GitHub
- Use the NVIDIA Forums
- Reach out on Slack
- Talk to NVIDIA Services

### Get Engaged

@RAPIDSai

https://github.com/rapidsai

https://rapids.ai/slack-invite/

https://rapids.ai