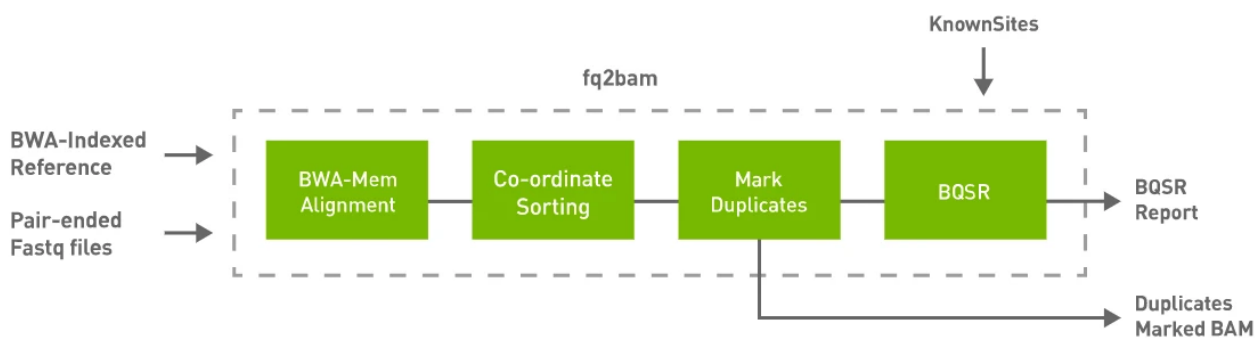


Accelerate FASTQ to BAM conversion using GPU and Parabricks



Refs to Parabricks: [fq2bam \(FQ2BAM + BWA-MEM\)](#)

Parabricks `fq2bam` is a software tool that can generate BAM/CRAM output from one or more pairs of FASTQ files. This tool takes advantage of the parallel computing capabilities of GPUs to speed up the analysis process.

To use Parabricks, users can provide input files in FASTQ format and specify the reference genome they wish to use for alignment. The software uses a proprietary algorithm to perform read alignment, variant calling, and quality control. The output is then generated in BAM or CRAM format, depending on the user's preference.

In this case study, we will align our sample and reference genome for further comparison and analysis. Following is our reference genome.

Preparation

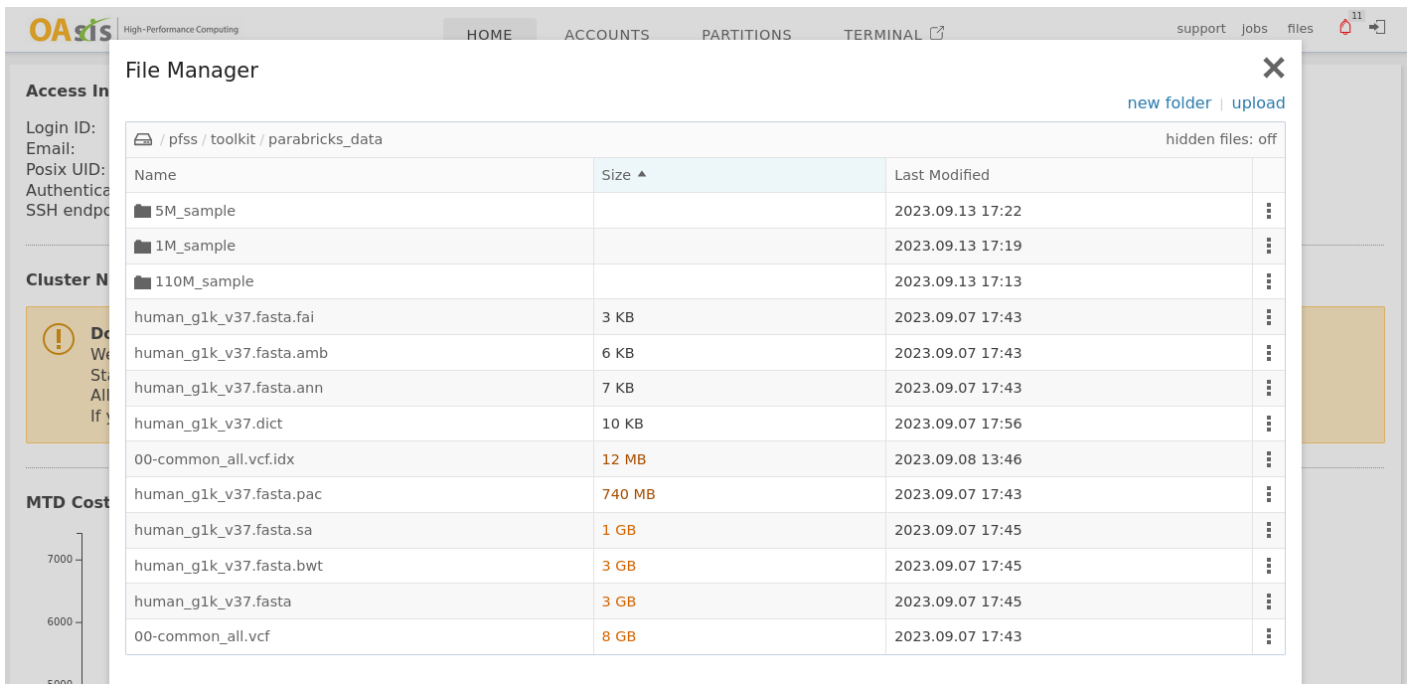
During conversion, `fq2bam` will align the sample with a reference genome. In this case study, we will utilize two human genomes. Following are the steps to download and index them.



Reference Genome: human_g1k_v37.fasta
Sample Data Source: SRA SRR7733443
Number Of Read: 2 x 5M bp
Read length: 150bp

1) Prepared data

OAsis comes with some prepared samples for you to start with. They are located at **/pfss/toolkit/parabricks_data**. Following is the folder structure.



Name	Size	Last Modified
5M_sample		2023.09.13 17:22
1M_sample		2023.09.13 17:19
110M_sample		2023.09.13 17:13
human_g1k_v37.fasta.fai	3 KB	2023.09.07 17:43
human_g1k_v37.fasta.amb	6 KB	2023.09.07 17:43
human_g1k_v37.fasta.ann	7 KB	2023.09.07 17:43
human_g1k_v37.dict	10 KB	2023.09.07 17:56
00-common_all.vcf.idx	12 MB	2023.09.08 13:46
human_g1k_v37.fasta.pac	740 MB	2023.09.07 17:43
human_g1k_v37.fasta.sa	1 GB	2023.09.07 17:45
human_g1k_v37.fasta.bwt	3 GB	2023.09.07 17:45
human_g1k_v37.fasta	3 GB	2023.09.07 17:45
00-common_all.vcf	8 GB	2023.09.07 17:43

human_* are the reference genomes and their index.

00-common_* are the well-known genomes and their index.

The three folders (5M_sample, 1M_sample, 110M_sample) are the testing samples in different sizes.

To utilize these prepared data, run the following command to copy them into your scratch folder.

The sample size is large. Please prepare about 100GB of free space. We suggest to use the scratch directory.

```
cp /pfss/toolkit/parabricks_data $SCRATCH/parabricks
```

2) Or download and index yourself

1. Download the sample genome for analysis.

- a. Download the SRA toolkit from
<https://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=software#header-global>
 - b. `tar xfv sratoolkit.2.10.5-centos_linux64.tar.gz`
 - c. `sratoolkit.2.10.5-centos_linux64/bin/fastq-dump -X 5000000 --split-files SRR9932168`
2. Download the reference genome.
- a. Download reference from
http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/reference/human_g1k_v37.fasta.gz
3. Index the genome.
- a. Leverage our Parabricks container, which already includes samtools.
 - b.

```
singularity run --nv /pfss/containers/clara-parabricks.4.0.0-1.sif bash  
samtools faidx ./human_g1k_v37.fasta
```
 - c. install bwa and use it to index human_g1k_v37.fasta
 - a.

```
# install bwa  
module load GCCcore/11.3.0 git/2.36.0-nodocs  
module load GCC/8.3.0  
git clone https://github.com/lh3/bwa.git  
cd bwa  
make  
  
# request a compute node to perform the indexing  
srun -p batch -c 32 --mem=100g --pty bash  
  
# in the prompt, run:  
./bwa index ../human_g1k_v37.fasta
```
4. Download Known Site
- a. download 00-common_all.vcf from
https://ftp.ncbi.nih.gov/snp/organisms/human_9606_b150_GRCh37p13/VCF/
 - b. `singularity run /pfss/containers/gatk.sif gatk IndexFeatureFile -I /00-common_all.vcf`

Convert sample from FASTQ to BAM

1) Using GPU with Parabricks

Parabricks does not support MIG yet. We will utilize one NVIDIA A100 here. We will use the prepared 5M_sample.

```
# request a GPU node with an A100
srun -p gpu -c 16 --gres=gpu:a100:1 --mem=128g --pty bash

# launch the Parabricks container
singularity run --nv /pfss/containers/clara-parabricks.4.1.0-1.sif bash

# inside the container, go to the folder we prepared in the last step
cd $SCRATCH/parabricks

# create a tmp folder which is needed by Parabricks
mkdir -p tmp

# launch the conversion process
NVIDIA_VISIBLE_DEVICES="" pbrun fq2bam \
  --num-gpus 1 \
  --ref ./human_g1k_v37.fasta \
  --in-fq ./5M_sample/SRR9932168_1.fastq ./5M_sample/SRR9932168_2.fastq \
  --out-bam ./mark_dups_gpu.bam \
  --tmp-dir ./tmp \
  --knownSites ./00-common_all.vcf \
  --out-recal-file ./recal_gpu.txt

# you may replace ./5M_sample/SRR9932168_1.fastq and ./5M_sample/SRR9932168_2.fastq with your own
samples
```

The output should look like the following:

```
[main] CMD: /usr/local/parabricks/binaries/bin/bwa mem -Z ./pbOpts.txt -F 0
/pfss/scratch02/appcara/parabricks_2/human_g1k_v37.fasta
/pfss/scratch02/appcara/parabricks_2/5M_sample/SRR9932168_1.fastq
/pfss/scratch02/appcara/parabricks_2/5M_sample/SRR9932168_2.fastq
@RG\tID:SRR9932168.1.1\tLB:lib1\tPL:bar\tSM:sample\tPU:SRR9932168.1.1
[main] Real time: 73.634 sec; CPU: 936.517 sec
[PB Info 2023-Sep-13 17:21:43] -----
[PB Info 2023-Sep-13 17:21:43] ||      Program:          GPU-BWA mem, Sorting Phase-I      ||
[PB Info 2023-Sep-13 17:21:43] ||      Version:              4.1.0-1      ||
[PB Info 2023-Sep-13 17:21:43] ||      Start Time:          Wed Sep 13 17:20:29 2023      ||
```

```
[PB Info 2023-Sep-13 17:21:43] ||      End Time:                Wed Sep 13 17:21:43 2023      ||
[PB Info 2023-Sep-13 17:21:43] ||      Total Time:                1 minute 14 seconds      ||
[PB Info 2023-Sep-13 17:21:43] -----
[PB Info 2023-Sep-13 17:21:43] -----
[PB Info 2023-Sep-13 17:21:43] ||      Parabricks accelerated Genomics Pipeline      ||
[PB Info 2023-Sep-13 17:21:43] ||      Version 4.1.0-1                ||
[PB Info 2023-Sep-13 17:21:43] ||      Sorting Phase-II                ||
[PB Info 2023-Sep-13 17:21:43] -----
[PB Info 2023-Sep-13 17:21:43] progressMeter - Percentage
[PB Info 2023-Sep-13 17:21:43] 0.0      0.00 GB
[PB Info 2023-Sep-13 17:21:48] Sorting and Marking: 5.000 seconds
[PB Info 2023-Sep-13 17:21:48] -----
[PB Info 2023-Sep-13 17:21:48] ||      Program:                Sorting Phase-II      ||
[PB Info 2023-Sep-13 17:21:48] ||      Version:                4.1.0-1      ||
[PB Info 2023-Sep-13 17:21:48] ||      Start Time:            Wed Sep 13 17:21:43 2023      ||
[PB Info 2023-Sep-13 17:21:48] ||      End Time:              Wed Sep 13 17:21:48 2023      ||
[PB Info 2023-Sep-13 17:21:48] ||      Total Time:            5 seconds      ||
[PB Info 2023-Sep-13 17:21:48] -----
[PB Info 2023-Sep-13 17:21:49] -----
[PB Info 2023-Sep-13 17:21:49] ||      Parabricks accelerated Genomics Pipeline      ||
[PB Info 2023-Sep-13 17:21:49] ||      Version 4.1.0-1                ||
[PB Info 2023-Sep-13 17:21:49] ||      Marking Duplicates, BQSR                ||
[PB Info 2023-Sep-13 17:21:49] -----
[PB Info 2023-Sep-13 17:21:49] BQSR using CUDA device(s): { 0 }
[PB Info 2023-Sep-13 17:21:49] Using PBBinBamFile for BAM writing
[PB Info 2023-Sep-13 17:21:49] progressMeter - Percentage
[PB Info 2023-Sep-13 17:21:59] 0.0      4.49 GB
[PB Info 2023-Sep-13 17:22:09] 0.0      4.49 GB
[PB Info 2023-Sep-13 17:22:19] 0.0      4.49 GB
[PB Info 2023-Sep-13 17:22:29] 0.0      4.49 GB
[PB Info 2023-Sep-13 17:22:39] 88.7      1.81 GB
[PB Info 2023-Sep-13 17:22:49] 100.0      0.00 GB
[PB Info 2023-Sep-13 17:22:49] BQSR and writing final BAM: 60.028 seconds
[PB Info 2023-Sep-13 17:22:49] -----
[PB Info 2023-Sep-13 17:22:49] ||      Program:                Marking Duplicates, BQSR      ||
[PB Info 2023-Sep-13 17:22:49] ||      Version:                4.1.0-1      ||
[PB Info 2023-Sep-13 17:22:49] ||      Start Time:            Wed Sep 13 17:21:49 2023      ||
[PB Info 2023-Sep-13 17:22:49] ||      End Time:              Wed Sep 13 17:22:49 2023      ||
[PB Info 2023-Sep-13 17:22:49] ||      Total Time:            1 minute 0 seconds      ||
[PB Info 2023-Sep-13 17:22:49] -----
```

The total time spent is 2m19s.

1) Using CPU with the compatible BWA-MEM, GATK4 Commands

```
# request CPU nodes
srun -p batch -c 32 --mem=100g --pty bash

# launch the container
singularity run /pfss/containers/gatk.4.4.0.0.sif bash

# go to the working folder and install bwa
cd $SCRATCH/parabricks
git clone https://github.com/lh3/bwa.git
cd bwa
make

# perform alignment
./bwa mem -t 32 -K 10000000 -R '@RG\tID:SRR9932168.1.1 \tLB:lib1\tPL:bar\tSM:sample\tPU:SRR9932168.1.1 ' \
../human_g1k_v37.fasta \
../SRR9932168_1.fastq \
../SRR9932168_2.fastq | \
gatk SortSam \
--java-options -Xmx30g \
--MAX_RECORDS_IN_RAM 5000000 \
-I /dev/stdin \
-O ../cpu.bam \
--SORT_ORDER coordinate

# for max spot id 5000000 spent 2.72 mins for sorting, 2.2 mins for convert to BAM
# [main] CMD: ./bwa mem -t 32 -K 10000000 -R @RG\tID:SRR9932168.1.1
\tLB:lib1\tPL:bar\tSM:sample\tPU:SRR9932168.1.1
/pfss/scratch02/appcara/parabricks/parabricks_sample/Ref/human_g1k_v37.fasta
/pfss/scratch02/appcara/parabricks/sratoolkit.3.0.6-centos_linux64/SRR9932168_1.fastq
/pfss/scratch02/appcara/parabricks/sratoolkit.3.0.6-centos_linux64/SRR9932168_2.fastq
# [main] Real time: 151.130 sec; CPU: 3594.553 sec
# INFO 2023-08-01 05:58:33 SortSam Finished reading inputs, merging and writing to output now.
# INFO 2023-08-01 05:58:47 SortSam Wrote 10,000,000 records from a sorting collection. Elapsed time:
00:02:43s. Time for last 10,000,000: 14s. Last read position: */*
# [Tue Aug 01 05:58:47 GMT 2023] picard.sam.SortSam done. Elapsed time: 2.72 minutes.
# Runtime.totalMemory()=1409286144
```

```

# Tool returned:
# 0

# generate .dict file
cd $SCRATCH/parabricks
gatk CreateSequenceDictionary -R human_g1k_v37.fasta

# mark duplicates (takes around 1 min)
gatk MarkDuplicates \
  --java-options -Xmx30g \
  -I ./cpu.bam \
  -O ./mark_dups_cpu.bam \
  -M metrics.txt

# generate a BQSR report
gatk IndexFeatureFile -I ./00-common_all.vcf

# recalibrate (takes around 1.68 mins)
gatk BaseRecalibrator \
  --java-options -Xmx30g \
  --input ./mark_dups_cpu.bam \
  --output ./recal_cpu.txt \
  --known-sites ./00-common_all.vcf \
  --reference ./human_g1k_v37.fasta

```

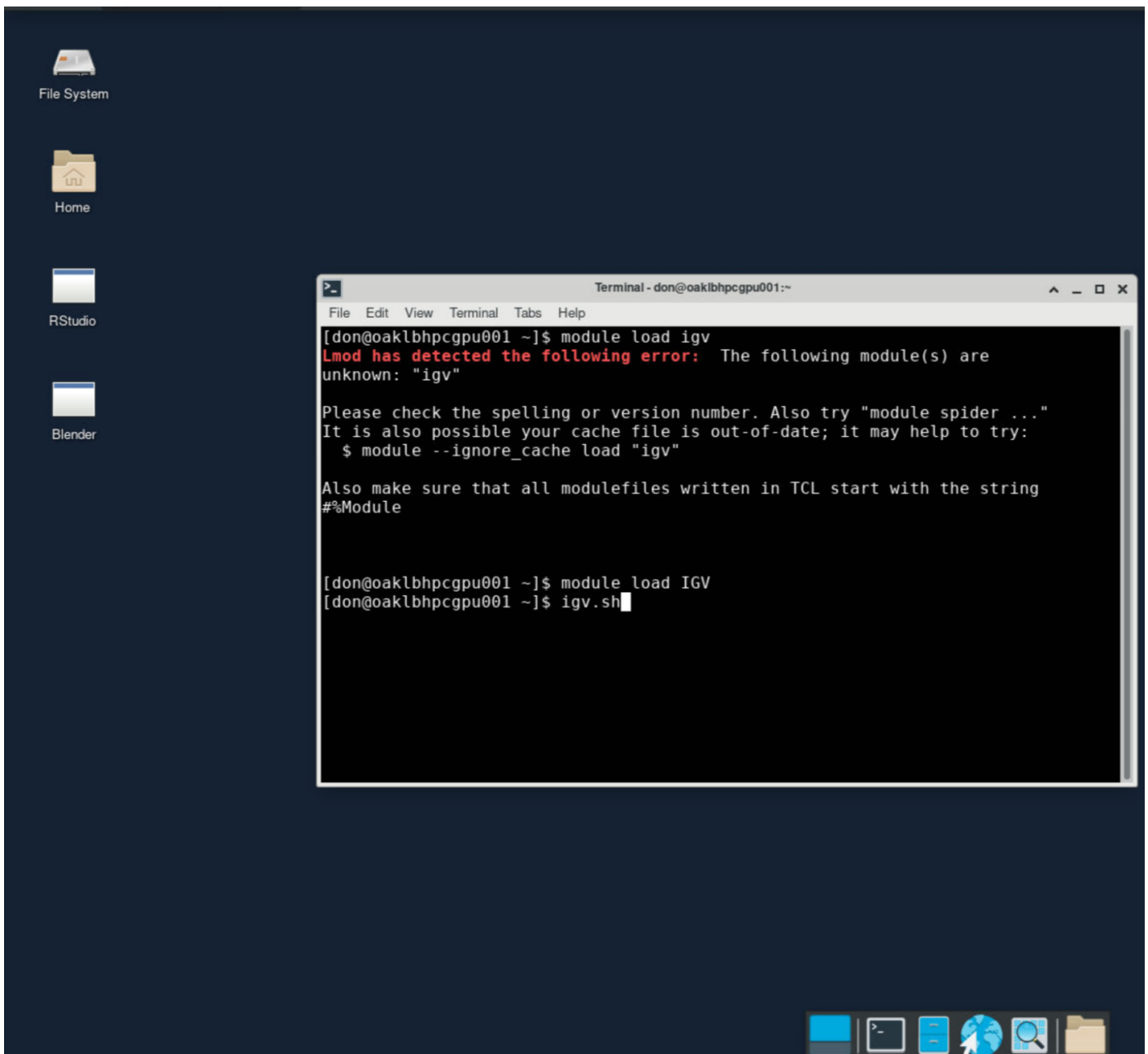
Performance comparison

Compute node \ Sample size (bp)	1M	5M	110M
32 core 100g mem	1.63 mins	7.56 mins	140.47 mins
16 core 128g mem A100 x 1	1.66 mins	2. 33mins	24.4 mins

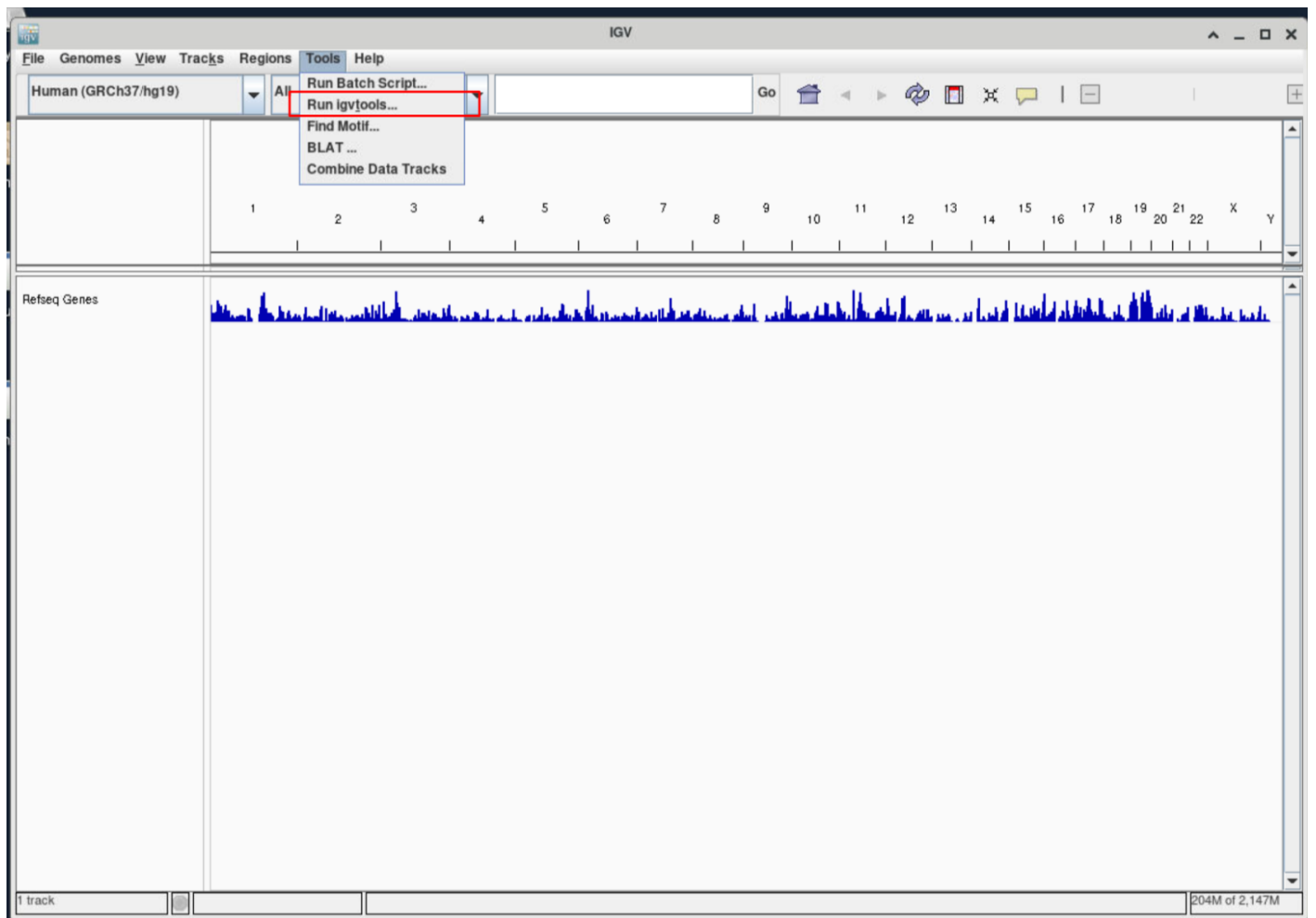
One of the main advantages of Parabricks is its speed. The software can analyze large datasets in a fraction of the time it would take traditional tools to complete the same task. Parabricks is also highly scalable and can analyze datasets of varying sizes without sacrificing performance. It is ideal for researchers and scientists who need to process large amounts of genomic data quickly and efficiently.

View bam file in IGV

After we get a BAM file, you may want to inspect it with a GUI tool like IGV. Oasis has pre-installed it as a module. First, request a VNC session from our web portal, load the IGV module in the terminal, then execute `igv.sh` to open IGV.



You can use `igvtools` to index the bam file.



FileGenomesViewTr

Human (GRCh37/hg19)

Refseq Genes

track

igvtools

CommandIndex

Input File /pfss/scratch02/appcara/parabricks_2/output/cpu.bam

Output File

Genome hg19

TDF and Count options

Zoom Levels

Window Functions

Probe to Loci Mapping

Window Size

Extension Factor

Count as Pairs

Min

Max

2%

10%

Absolute Max

90%

Mean

98%

Median

Browse

Sort Options

Temp Directory

Max Records

Close

Run

Messages

5000000 reads processed ...

6000000 reads processed ...

7000000 reads processed ...

8000000 reads processed ...

9000000 reads processed ...

10000000 reads processed ...

11000000 reads processed ...

12000000 reads processed ...

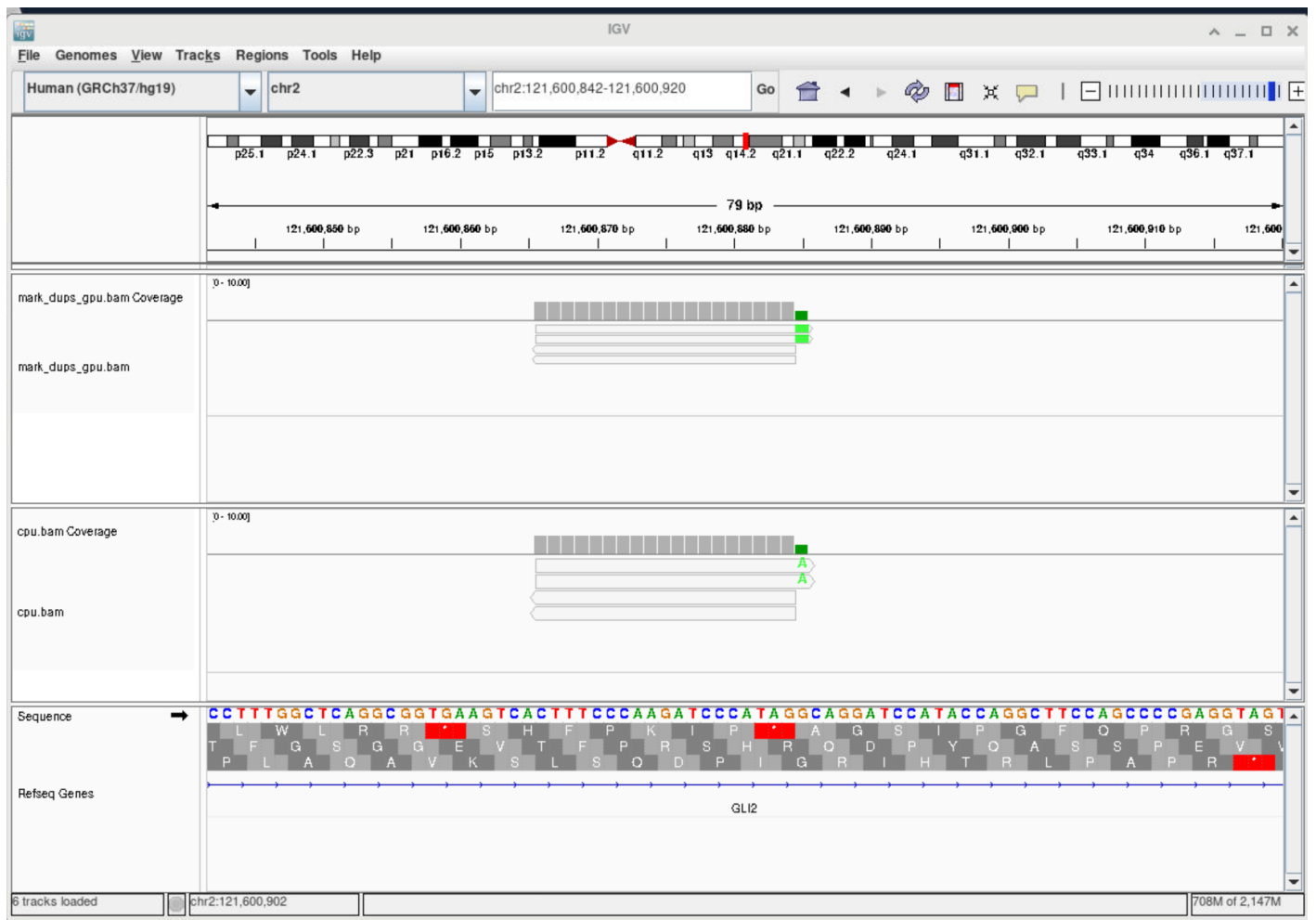
13000000 reads processed ...

X

Y

I

1,210M of 2,147M



Revision #37

Created 3 August 2023 01:42:48 by Don Chu

Updated 14 September 2023 02:31:51 by Don Chu