Getting started with whole genome mapping and variant calling on the command line

Georgina Samaha
Sydney Informatics Hub
University of Sydney

Australian BioCommons

Hello :)

- Bioinformatician at Sydney Informatics Hub, University of Sydney
- Australian BioCommons **‘Bring Your Own Data’-CLI project**
- My research interests: heritable diseases in domestic cats, conservation genomics

Today

A framework for approaching mapping and variant calling workflows on the command line.

What does a typical workflow look like?

What can this look like in practice?

What workflow do I need for my experiment?

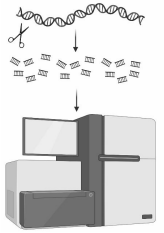
What are some existing pipelines I can use?

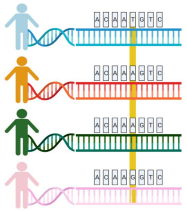
Where can I find accessible compute?

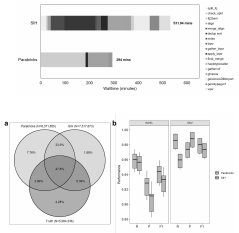
Australian BioCommons

Mapping and variant calling* basics

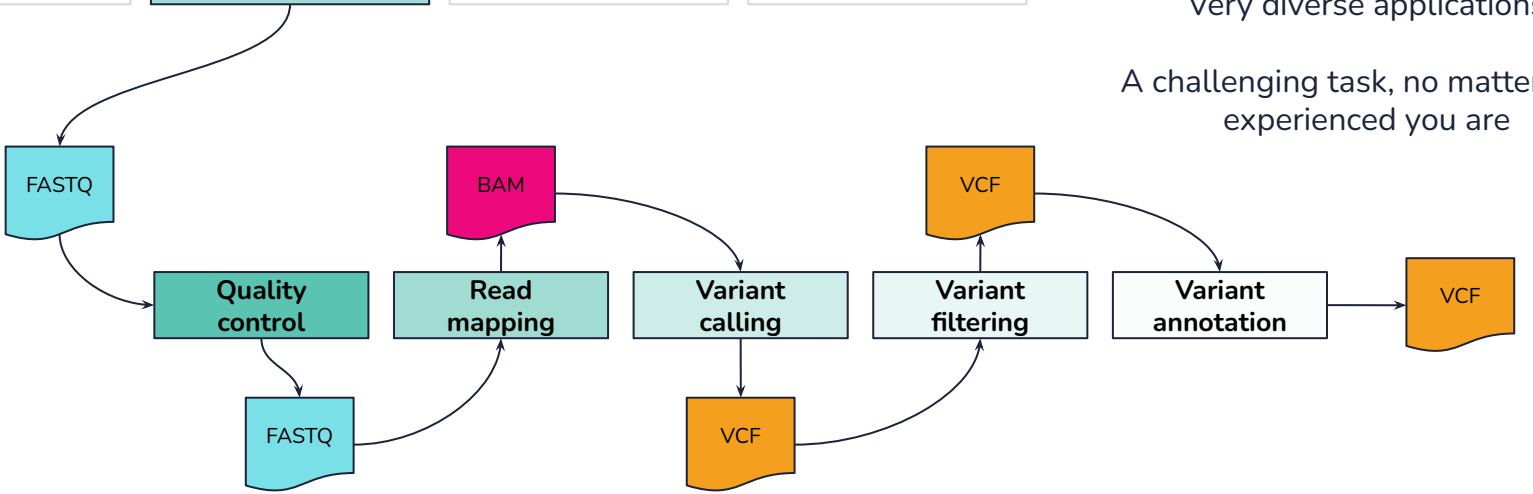
(* germline SNV and indel discovery with short-reads)







Data sequencing | **Data preprocessing** | Analysis | Interpretation



Process of aligning raw NGS reads to a reference genome and identifying variant sites

Multiple processes and tools

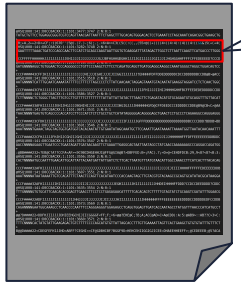
Computationally intensive

Very diverse applications

A challenging task, no matter how experienced you are

Australian BioCommons

The file formats - FASTQ (raw sequence reads)

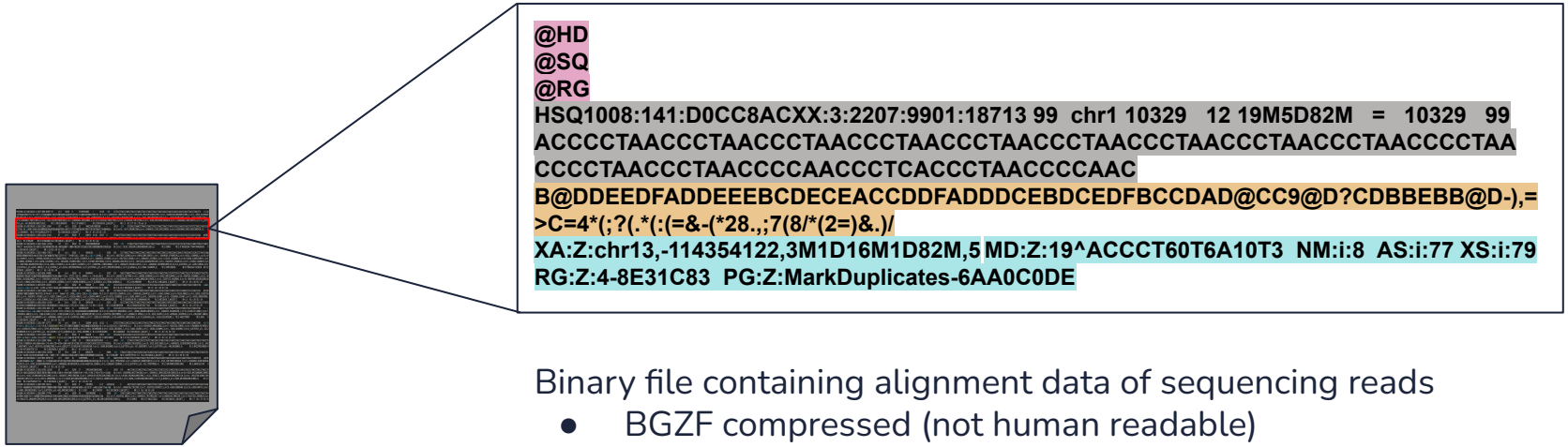


```
@HSQ1008:141:D0CC8ACXX:1:1101:1226:3541 2:N:0:1
CATTNNNNCTGCATATCAGAGACTTTTAGTACTTTCACTAATTTACTACTGCTATCTAGAAGGCAG
+
CCCF####24CFHIJJJJJHIIJIIJJJJGIJJJJHIIIIJJJIGIJIIJIJIJJJJIIIIJGIIGGJEHIJJHHHHHHHHFFFF
```

Text file containing sequence data

- Usually compressed (.fastq.gz or .fq.gz)
- **Header:** sequence identifier, info about the sequencing run
- **Read:** bases called as A, C, T, G, and N
- **Separator:** just a plus (+) sign
- **Per-base quality score:** phred +33 encoded, ASCII characters representing a numerical score

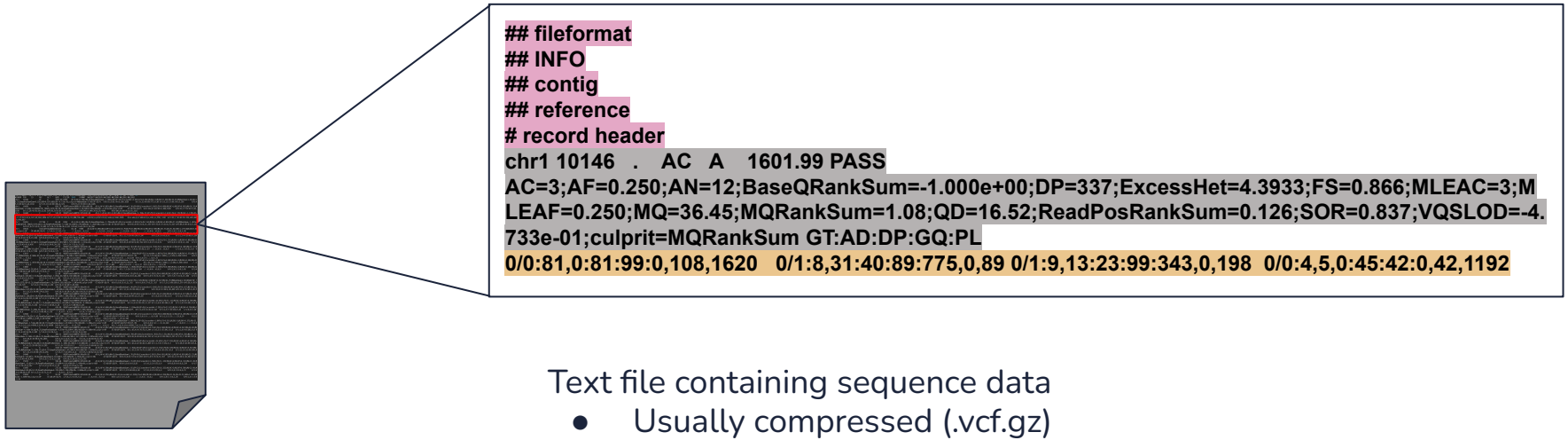
The file formats - BAM (aligned sequence reads)



Binary file containing alignment data of sequencing reads

- BGZF compressed (not human readable)
- **Header:** lines starting with @. Info about the file
- **Read:** read mapping info, bases called as A, C, T, G, and N
- **Mapping quality score:** phred +33 encoded, ASCII characters representing a numerical score
- **Tags:** Additional template and mapping data

The file formats - VCF (variant call details)



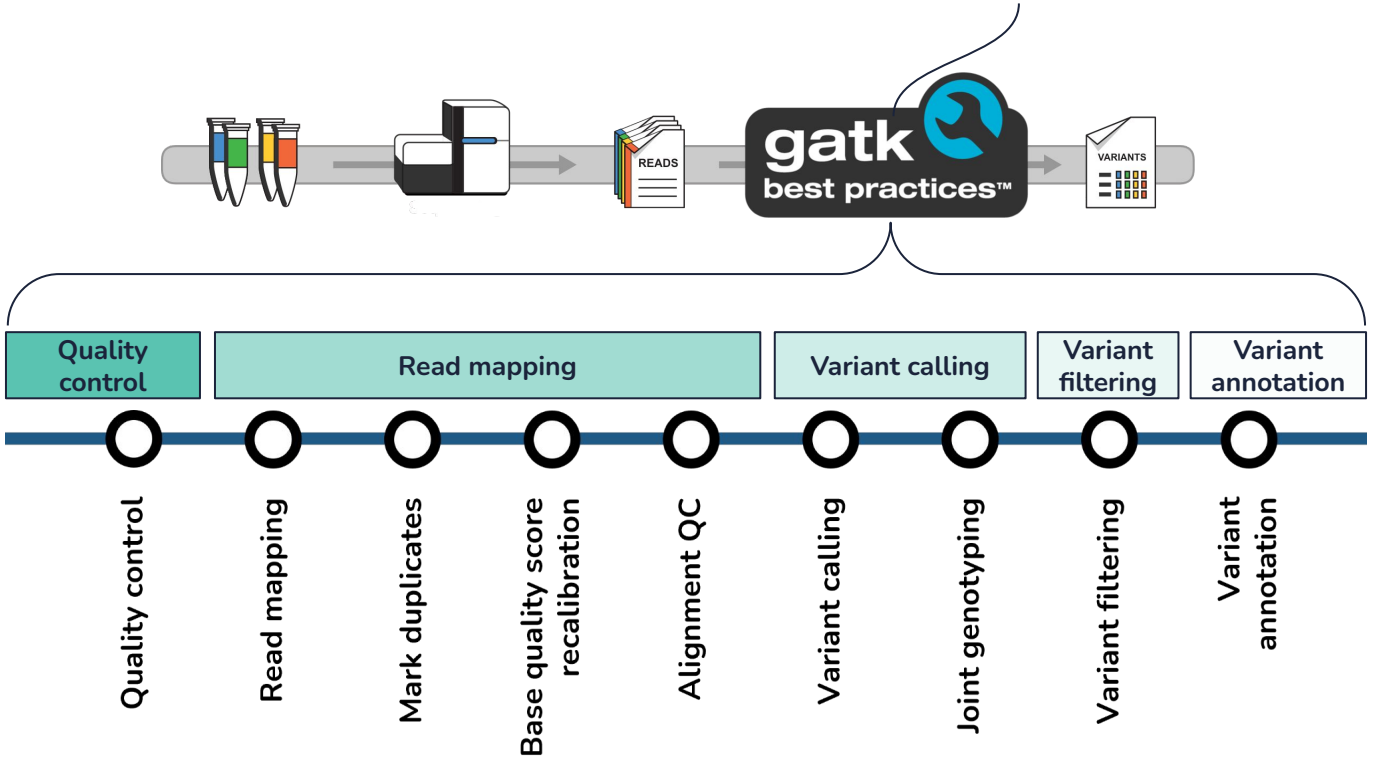
Text file containing sequence data

- Usually compressed (.vcf.gz)
- **Header:** variant calling, dataset, reference information
- **Variant site record:** location, alleles, quality, filters
- **Genotypes:** per-sample genotype, quality annotations

Australian BioCommons

The workflow

* Developed for humans, but commonly adapted for many other organisms.

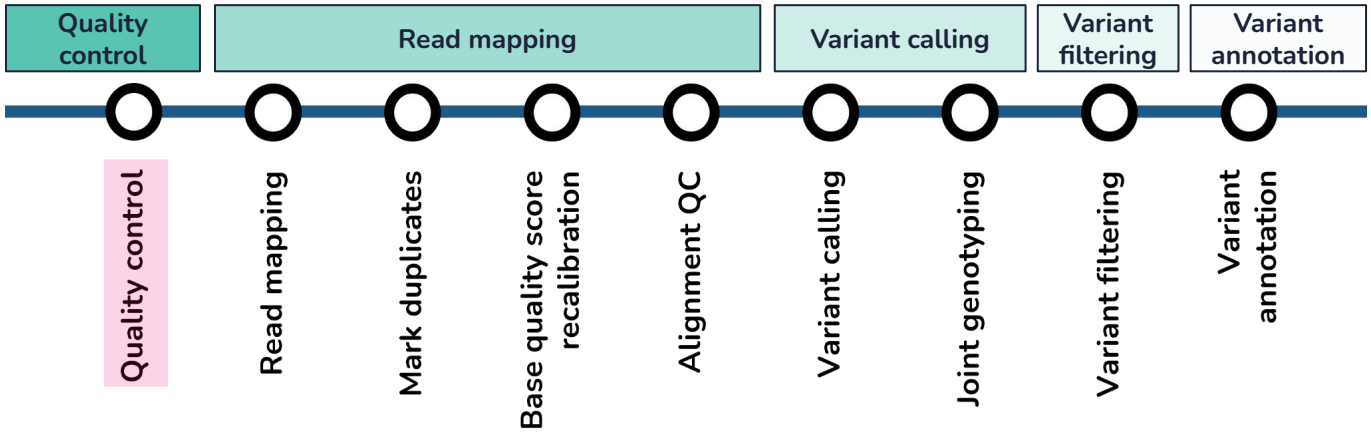


Australian BioCommons

1. Raw sequence QC

Short read sequencing is error prone. These errors can be indistinguishable from variants once they're processed.

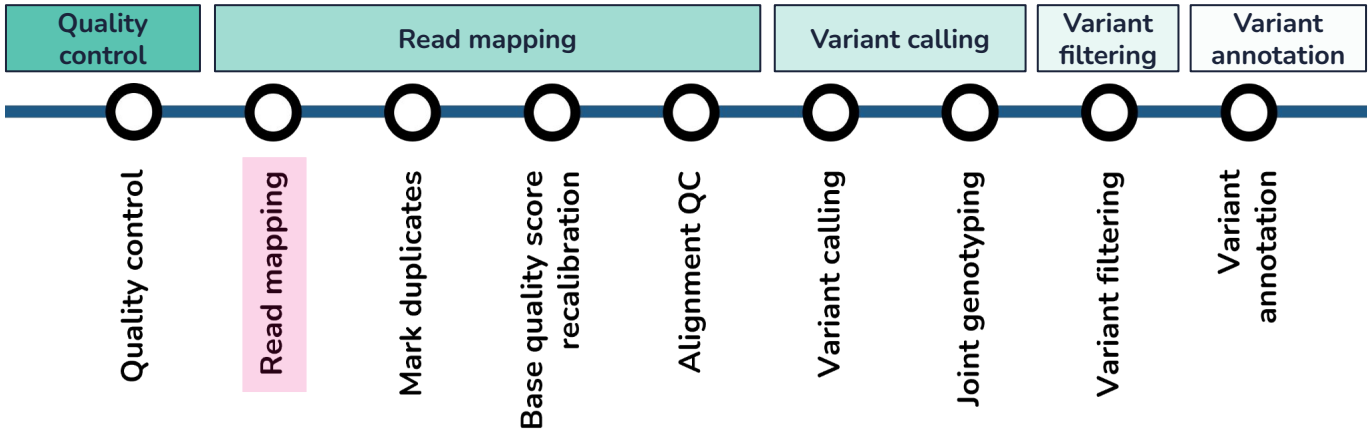
Quality checks and data cleaning at the outset saves time downstream.



2. Mapping reads to a reference genome

Process of determining the genomic position of all reads across a reference genome.

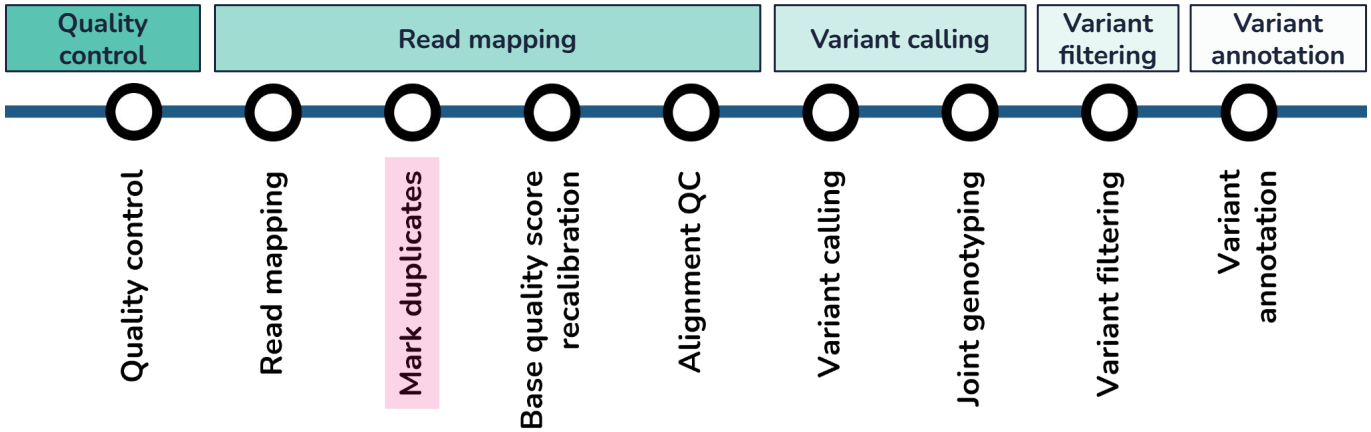
Bases in the reference are covered by multiple reads and the ‘depth’ of coverage can determine whether variant discovery can be accurately performed at a particular base.



Australian BioCommons

3. Mark duplicate reads

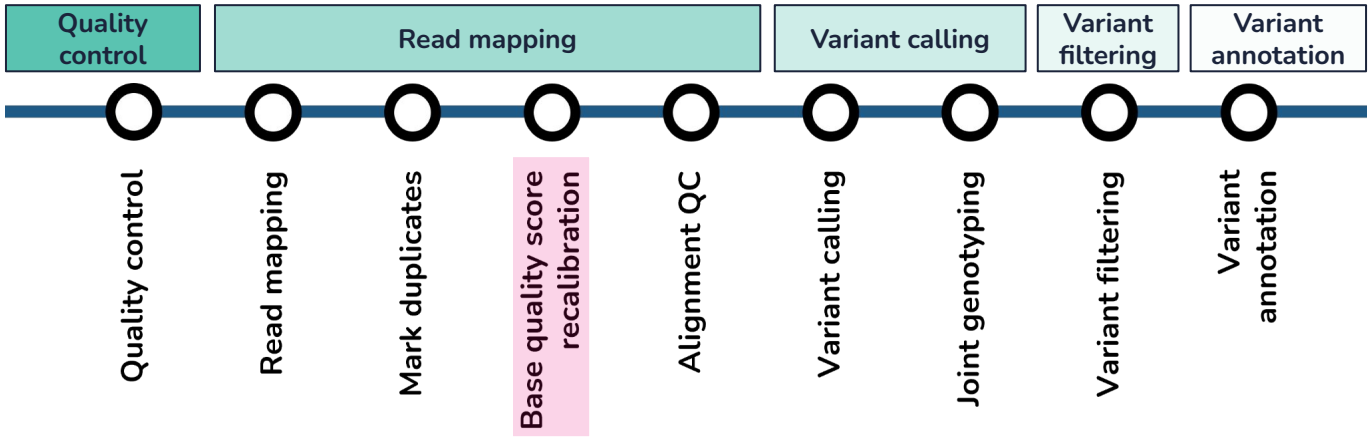
Duplicate reads that originate from the same strand of DNA can result in over-representation of areas preferentially amplified during sequencing. Marking these reads allows downstream tools to handle them properly.



4. Base quality score recalibration

Process of empirically modelling of the potential sequencing error profile of bases using known variants in the population and adjusting the scores accordingly.

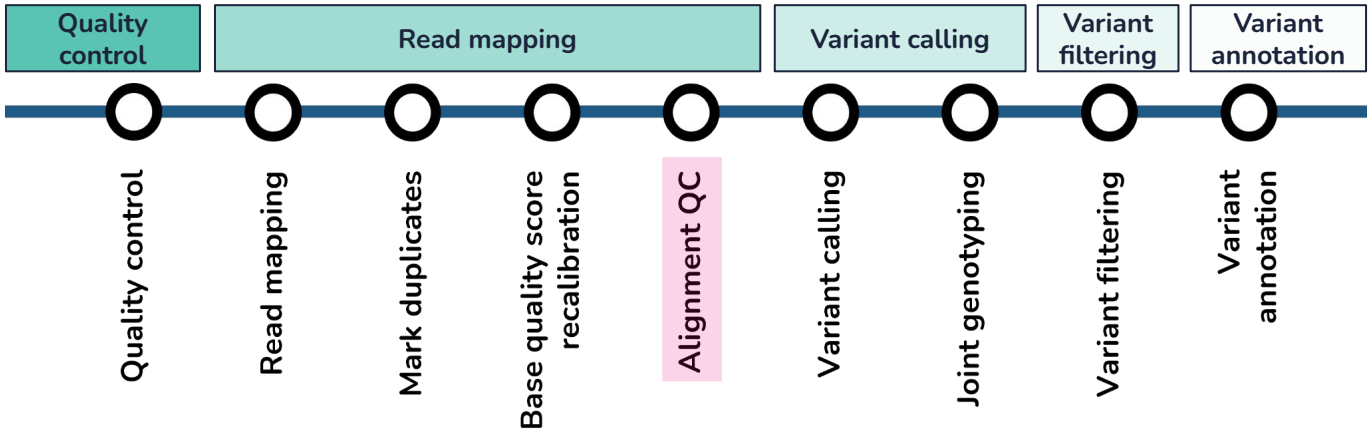
Variant calling algorithms rely on quality scores assigned to individual bases. Recalibrated quality scores improve variant calling accuracy.



Australian BioCommons

5. Alignment QC

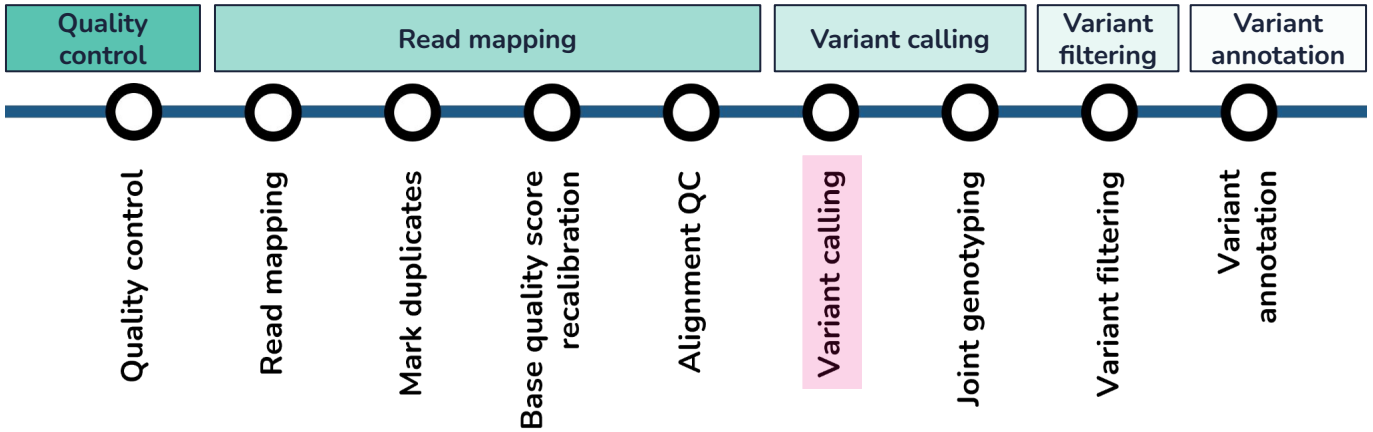
Collecting descriptive statistics for read coverage across the genome of a sample, mapped read lengths and mapping quality summarise the quality of output alignment files.



6. Identifying variant sites against the reference genome

Process of identifying variants by comparing the mapped sequences to the reference sequence in the genome for a sample.

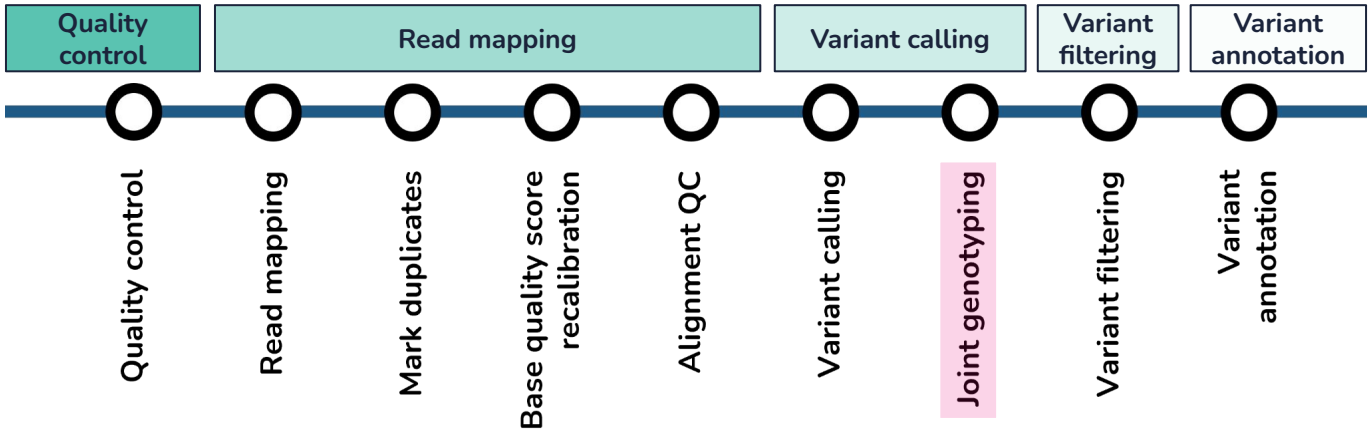
Variant calling tools and workflows have varying levels of sensitivity and specificity.



Australian BioCommons

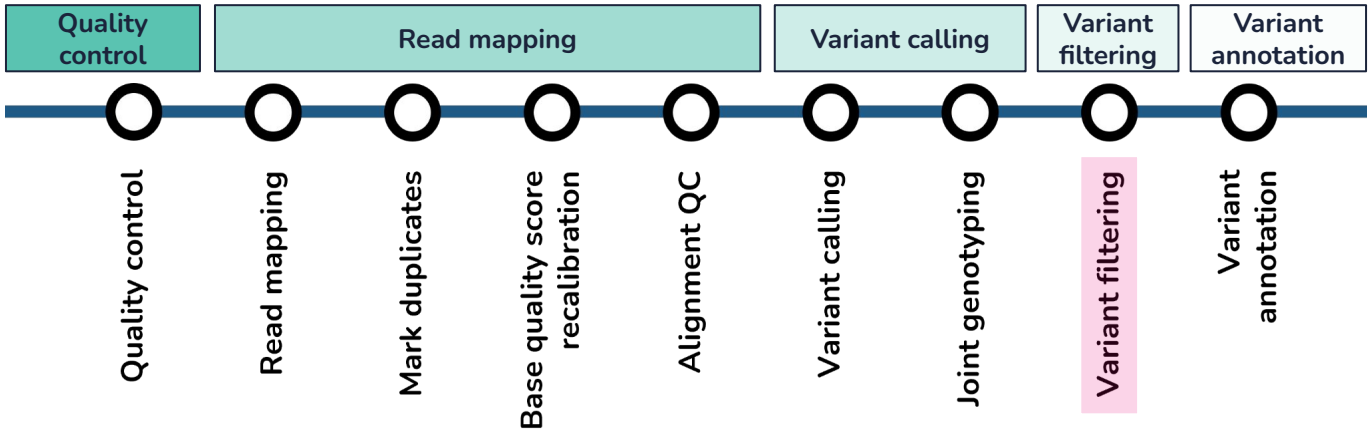
7. Joint genotyping of samples in a cohort

The joint analysis of multiple samples to produce a set of variants. This process can improve the sensitivity and accuracy of variant detection by leveraging cohort-wide information from multiple samples. It also makes it easier to compare across samples in downstream analyses.



8. Removal of low-confidence variants

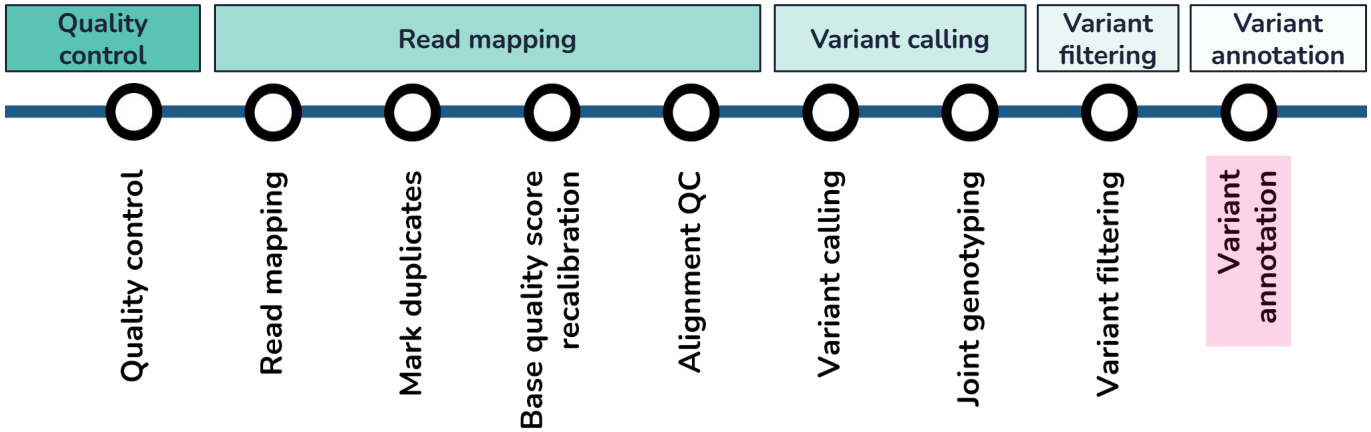
There are multiple approaches for variant filtering including variant quality score recalibration which uses genomic context to empirically re-define quality scores, and hard filtering which use strict quality thresholds.



9. Annotation of final variant set

The process of determining the potential functional impacts of a variant on a gene and its transcript and protein products.

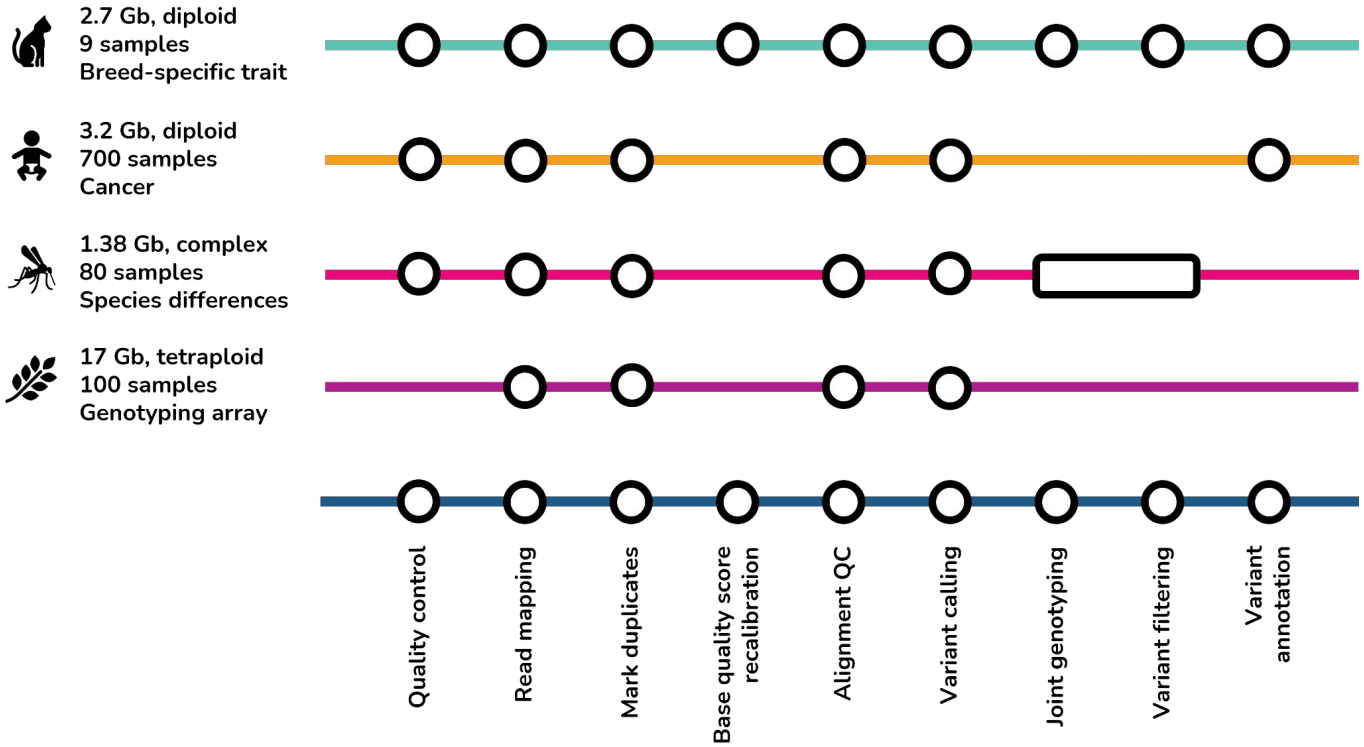
It is required for linking variants with phenotypic changes. We can also determine how many of our variants are known to exist in the population, and how prevalent they are.



Australian BioCommons

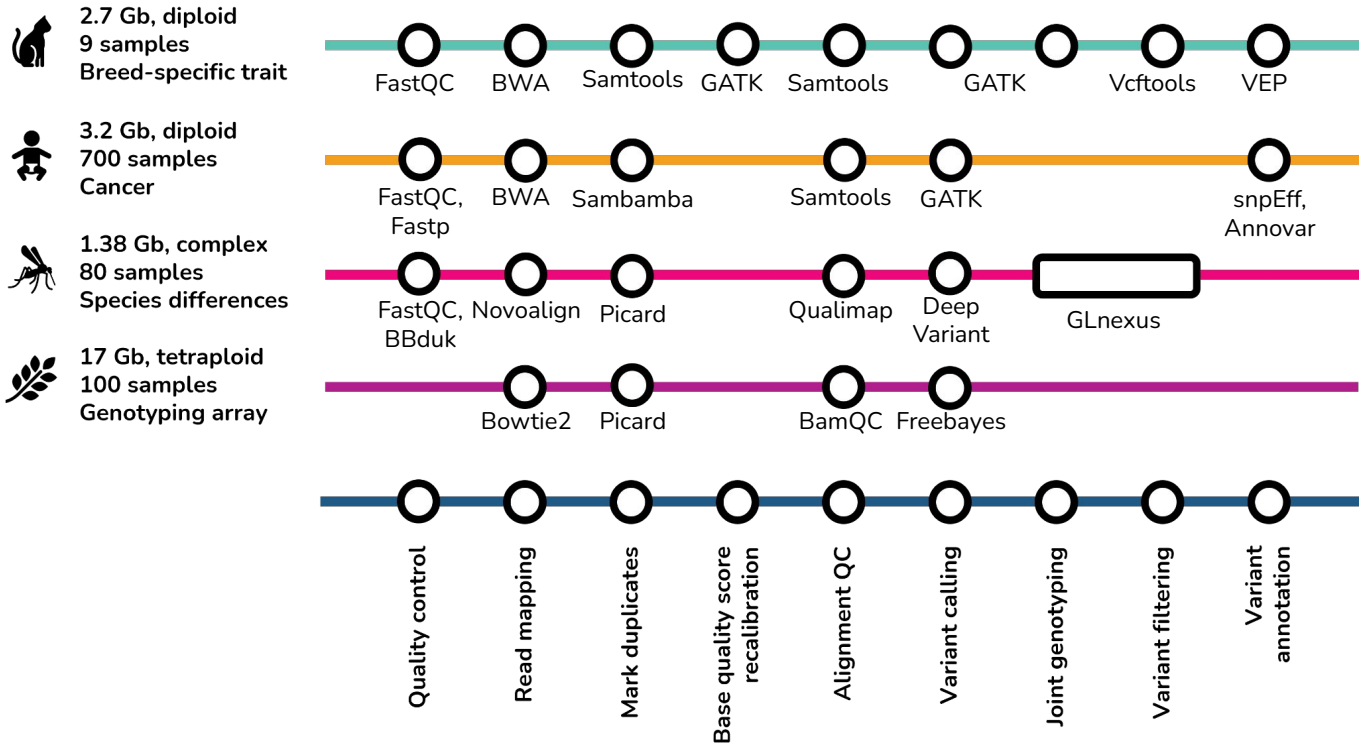
What can this workflow look like in practice?

Varied workflow design for different research questions



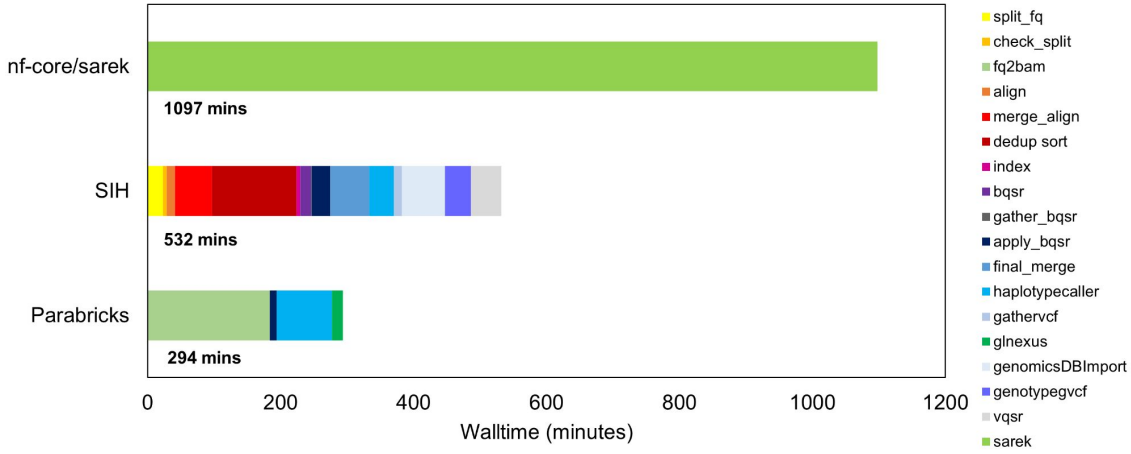
Australian BioCommons

Varied tool choices for different user requirements



Australian BioCommons

Varied user experiences



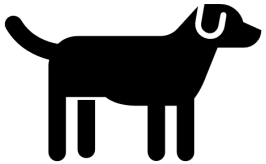
If you picked up each pipeline, followed the documentation with default parameters, how does it perform with processing 6 human samples?

DOI: 10.5281/zenodo.6930813

Australian BioCommons

How to choose the right pipeline for you?

Example project: heritable diseases in dogs



Australian BioCommons

Project summary

Research context

- Heritability of some diseases is breed-based
- Large multi-breed cohorts can be used for validation in various studies

Need to be consistent with existing community 'best practices'

Dataset

- Illumina, short read, paired-end WGS data
- 110 canine samples
- Samples collected for various studies

~3Tb of input data! Will need specialised compute and a scalable workflow.

Need a workflow that is flexible with input data requirements.

Bioinformatics

- Realign samples to new reference genome canFam4 and call variants
- Joint-called cohort VCF output requested
- Fast turnaround time needed for a publication

Requires high capacity HPC. Limited time for workflow development

This process is CPU and memory intensive at this scale.

Compute resources

- Has access to institutional HPC but its not capable of this scale
- Fastqs stored on institutional data store

Scratch working space limitations. Need alternative compute for input and output data.

What best practice guidelines should I follow?

Go to the literature. What's the current standard?

- GATK Best Practices: includes joint genotyping

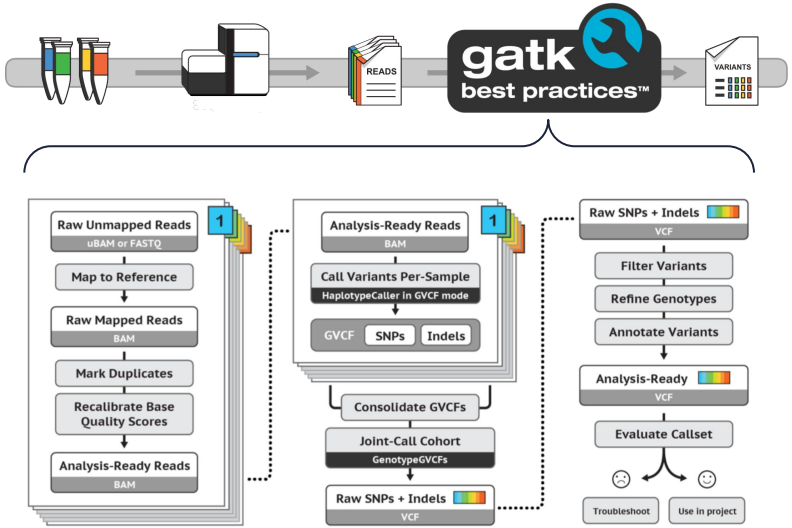
What answers does this give us?

✅ Base workflow structure

✅ Preferred tools for each step

Some considerations...

- **Depends on research Q**
- Can be **high-level** guidelines **or specific** to an application. (i.e. clinical, diagnostics)
- **Beware of legacy code**: what is 'best practice' changes over time



Australian BioCommons

How is my dataset structured?

How big is the input/output data?

- ~3 Tb raw reads (.fastq.gz)
- + ~3.1 Tb aligned reads (.bam)
- + ? Joint called variants (.vcf.gz)
- + ??? intermediate files

What reference materials are needed?

- ~2.5 Gb canFam4 reference (.fasta)
- Known canine variants (.vcf.gz)

How complex is the dataset?

- Diploid genome, ~2.5Gb
- Similar requirements to human genome

What does this answer?

✅ Workflow design
✅ Computing requirement estimates
✅ Need for flexibility/customisation of workflow

Some considerations...

- Not all organisms have **access to high quality reference genomes** and known variant data
- The larger and more complex the dataset, the more computing power needed
- **Don't forget the intermediate files!** They take up a lot of space

What are my user experience needs?

A workflow that can successfully be run

- Bioinformatics at HPC-CLI experience
- Pipeline optimised for high throughput

Need to customise

- GATK best practices are suitable for diploid, model species
- Variant quality score recalibration might need fine tuning for dogs

What does this answer?

✅ Need for workflow customisation
✅ CLI/coding experience level
✅ Type of computing infrastructure

Some considerations...

- If you are unsure, it is **safe to run tools with default parameters**
- The more broadly applicable a pipeline is, the less accurate it'll be
- **Specialised expertise** is required to efficiently use HPCs

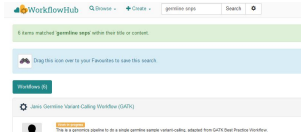
What have we got so far

- ✅ Base workflow structure - GATK best practices
- ✅ Limited customisation needs - model species, basic research question
- ✅ National HPC infrastructure - for high throughput
- ✅ No requirement for easy CLI

Where can I find existing workflows?

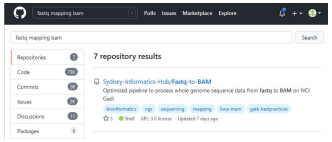
WorkflowFinder

- Searchable listing of installed and optimised workflows at Australian facilities



WorkflowHub

- Registry of published, citable scientific computational workflows

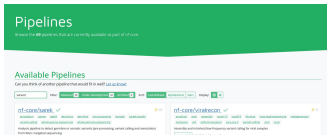


GitHub

- Search repositories with key words
- Published code from bioinformaticians and research organisations.

Ask a bioinformatician

- Bioinformatics core facilities
- Colleagues and collaborators
- Methods section of genomics papers



nf-core

- Community-curated set of Nextflow pipelines
- Reproducible, easy to deploy

Australian BioCommons

What are some existing pipelines I can use?

nf-core/sarek

Community curated, Nextflow implementation of best practice pipelines

+ Easy to install and run
+ Highly flexible
- Not optimised for scalability
- No joint genotyping option

NVIDIA PARABRICKS

Turnkey, GPU-accelerated bioinformatics tool suite

+ Easy to install and run
+ Optimised for scalability
- Commercially licensed
- Requires GPU hardware

Build your own custom pipeline

+ Highly customisable
- Long development timeline
- Time to optimise
- Manual tool installation

THE UNIVERSITY OF SYDNEY NCI AUSTRALIA

Series of bash scripts, optimised for efficiency and scalability at NCI Gadi

+ Optimised for scale
+ Meets NCI's efficiency standard
+ Highly flexible
- Manual run of each step

What is bioinformatics? → I'm a bioinformatician

Australian BioCommons

Where to find accessible compute?

Accessible computing for Australian life scientists

What facilities?

- Institutional HPCs
- Australian research computing facilities
    - HPCs, cloud computing, virtual machines

Pawsey NCI AUSTRALIA ARDC Nectar Research Cloud QCIF

    - Web-based bioinformatics platforms

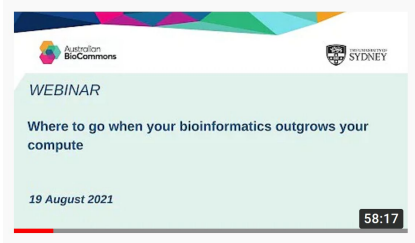
Galaxy AUSTRALIA

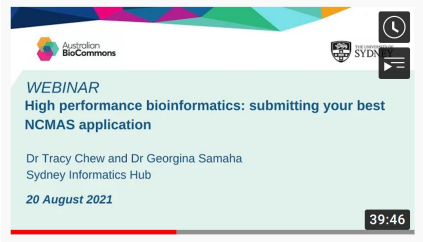
How do I get access?

- Contact your institutions ICT or eResearch services
- Merit schemes ADAPTER NCI NCMAS
- Start up, industry, institutional schemes
- Funded options

Compute webinars

DOI: 10.5281/zenodo.5240578





Australian BioCommons

Takeaways

A few takeaways

- ✅ GATK best practices are widely applicable across a lot of organisms and projects
- ✅ Following best practices are essential for reproducible genomics projects
- ✅ There is no ‘one size fits all’ solution for whole genome mapping and variant calling
- ✅ Ask questions about your data and experiment to find the solution that best meets your needs
- ✅ Existing pipelines can be suitable for different CLI experience levels, customisation needs
- ✅ National compute is available, contact your ICT service provider or core research facilities for support

Thank you :)