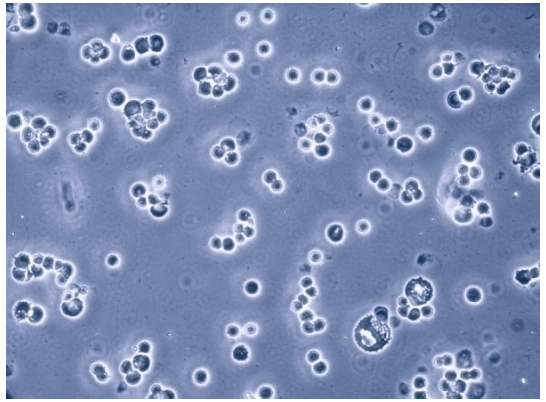
Sequencing pipelines

NeSI workshop, 11 August 2021

Dr Andrew Wallace

Gathering the data

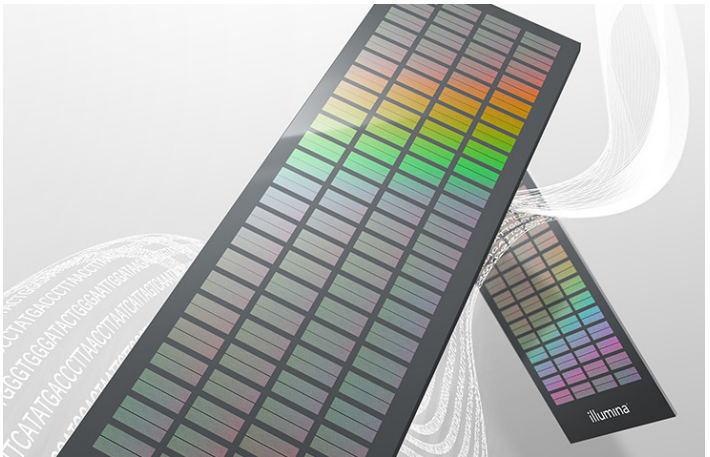
Genetic material



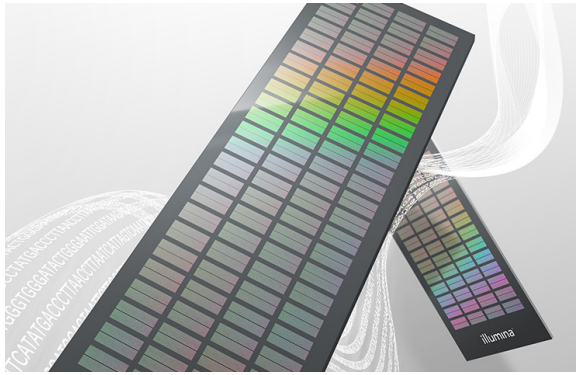
Extraction & library prep

Sequencing

Genotyping



Processing the data



Imputing GT: Beagle 5.0

Limitations:
- Need pre-made SNPchips
- Can't discover new variants

Identification: Kraken2

Mapping: BWA

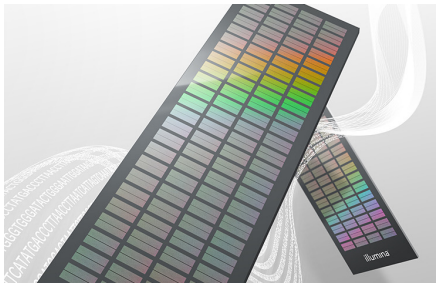
Variant calling: BCFtools

Imputing GL: Beagle 4.1

Imputing GT: Beagle 5.0

Different types of output data

Genotyping



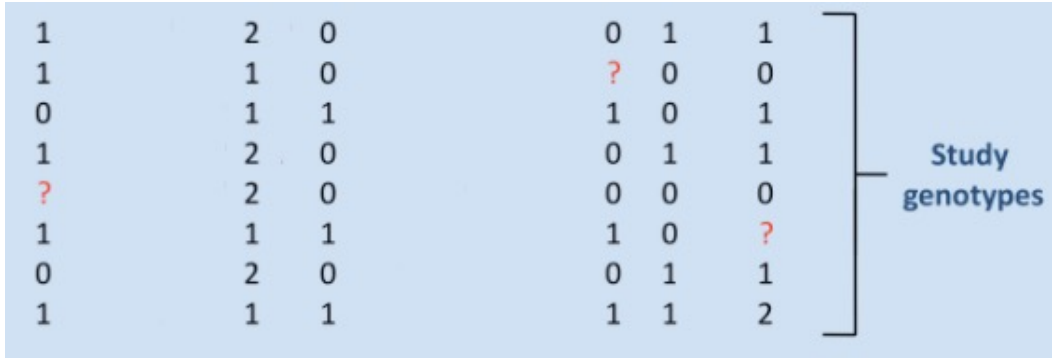
1	2	0	0	1	1	Study genotypes
1	1	0	?	0	0	
0	1	1	1	0	1	
1	2	0	0	1	1	
?	2	0	0	0	0	
1	1	1	1	0	?	
0	2	0	0	1	1	
1	1	1	1	1	2	

Sequencing

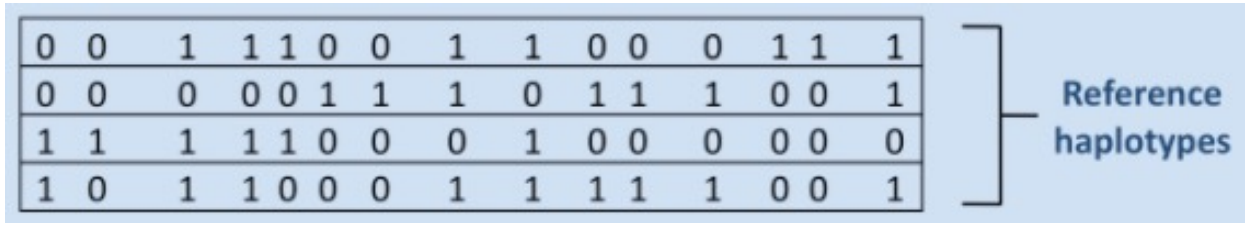
Reads:

ACACATACGCACTGGGCGTAAAGGGCGCGAGGCGTCGGTGCTC
AAAGTCC…
TTGTTAGTCGAGTGTGAAAGCCCTGGGCTTAACCCGGGAAGCG
CGTCAGT…
AAGGTCACGTGCAGGGCCTTAACGTGCAGTCCAGGGCAATTTC
GAGCTCT…
CGCGTTAATTGCCAGTTGCAGAACGTCCGTTACGTAAACGGTTT
TACGTC…
GGGCATGCATGCACGTACGTTGCAACCGTAGGAAGCCTTGACG

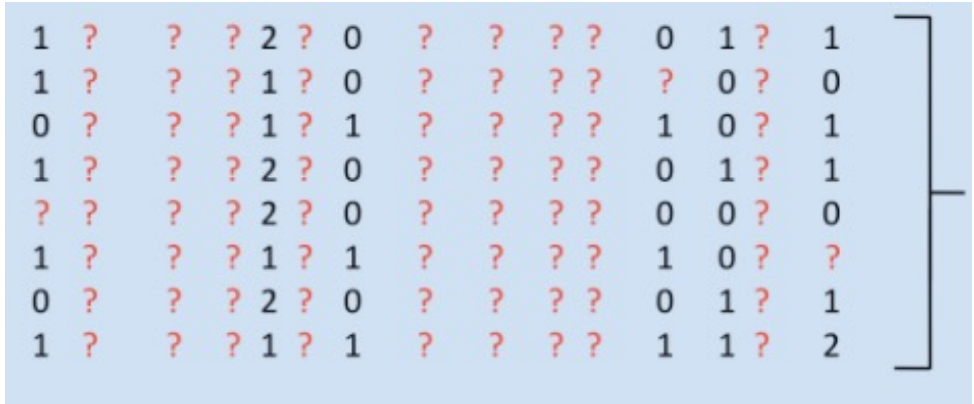
Imputation is guessing the blanks



Using known data



Infer



- An imputation program will statistically predict the most likely missing data
- The prediction can only be as good as the information you provide
- The 'Reference' set needs to be statistically representative of the population

The sequencing pipeline looks a bit different

Reads:

ACACATACGCACTGGGCGTAAAGGGCGCGAGGCGTCGGTGCTC
AAAGTCC…
TTGTTAGTCGAGTGTGAAAGCCCTGGGCTTAACCCGGGAAGCG
CGTCAGT…
AAGGTCACGTGCAGGGCCTTAACGTGCAGTCCAGGGCAATTTC
GAGCTCT…
CGCGTTAATTGCCAGTTGCAGAACGTCCGTTACGTAAACGGTTT
TACGTC…

Mapping to known reference genome

**Calling variants,
where sample differs to reference**

- Sequencing gives random tiny pieces of the genome
- Computer has to do the jigsaw

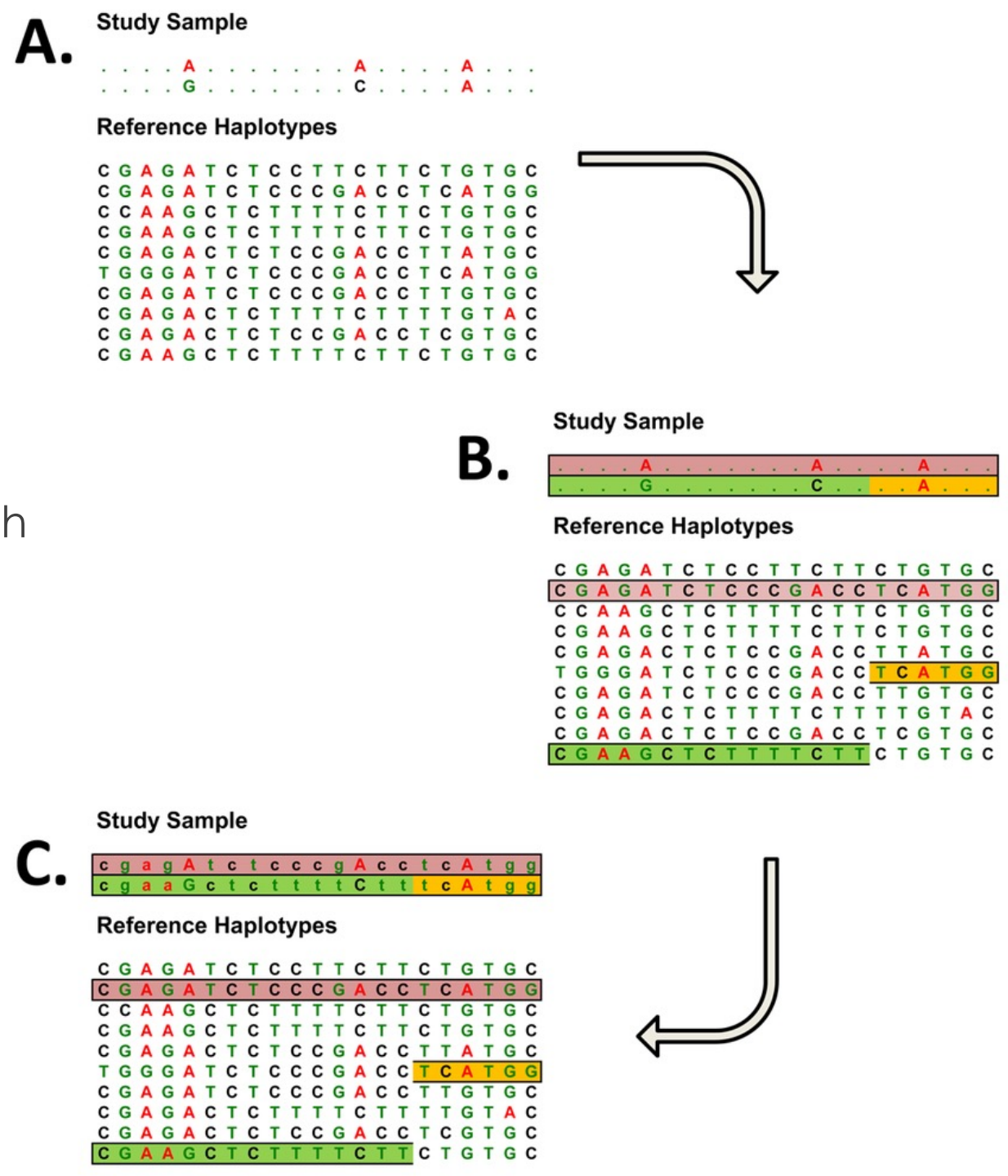
  'Mapping' or 'Assembly'

- Identify places where the samples differ

  'Variant calling'

The Imputation step

- Imputation program:
  - Finds the best match(es) in the reference for the sample sequence
  - Inserts the rest of the reference match into the sample



Typical process for low-pass sequencing

- Data arrives in **.fastq.gz** format
  - Readable text, zipped
  - Lines have ID of read, estimated accuracy, DNA read data in pieces
  - Typically millions of reads per sample
- Quality check
  - ***FastQC*** on NeSI
    - Analyses each **.fastq.gz** and generates a **.html** report on quality of sequence
    - Takes a few minutes per sample
    - Don't panic just because some FastQC metrics are bad

Raw read data:

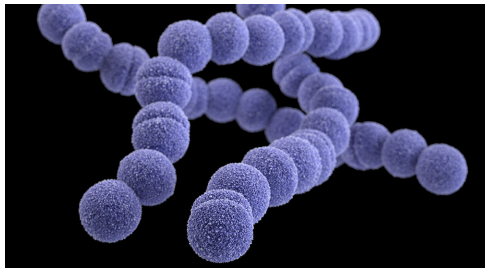
ACACATACGCACTGGGCGTAAAGGGCGCGAGGCGTCGGTGCTC
AAAGTCC…
TTGTTAGTCGAGTGTGAAAGCCCTGGGCTTAACCCGGGAAGCGC
GTCAGT…
AAGGTCACGTGCAGGGCCTTAACGTGCAGTCCAGGGCAATTTCG
AGCTCT…
CGCGTTAATTGCCAGTTGCAGAACGTCCGTTACGTAAACGGTTTT
ACGTC…
GGGCATGCATGCACGTACGTTGCAACCGTAGGAAGCCTTGACGT
TACGTT…

Identifying reads

- Identification (optional extra step)
  - ***Kraken2*** on NeSI can take an hour or two per sample
    - Requires pre-building a Kraken2 database, by downloading genomes from international sequence databases
    - Kraken2 can match reads in **.fastq.gz** files against its database and report what organism they match to
    - Generates text output for each fastq

```
9566081 9566081                                                  Bos taurus
```

```
4387    30              Terrabacteria group
3355    55                Firmicutes
3271    120                 Bacilli
3117    353                   Lactobacillales
2578    152                     Streptococcaceae
2295    1036                      Streptococcus
1078    1059                        Streptococcus suis
```



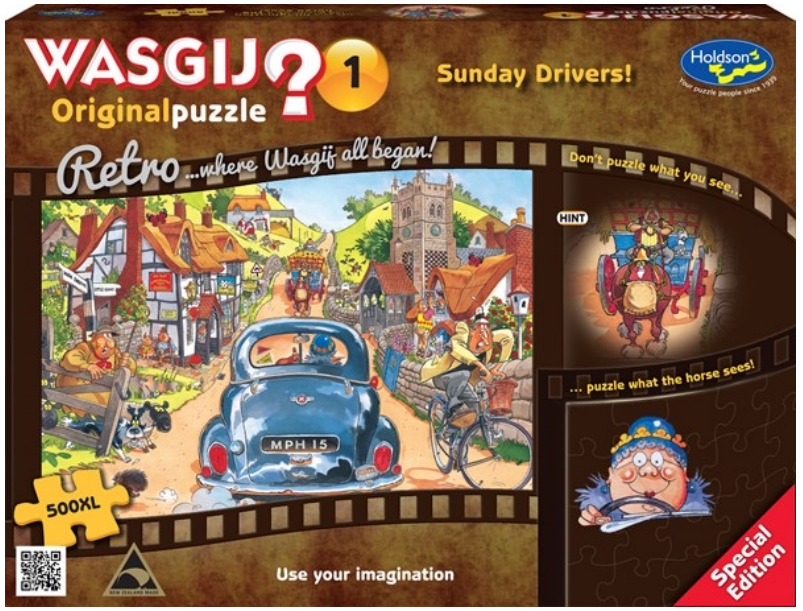
```
23      0               Archaea
23      0                 Euryarchaeota
23      0                   Methanomada group
23      0                     Methanobacteria
23      0                       Methanobacteriales
23      0                         Methanobacteriaceae
23      10                          Methanobrevibacter
11      11                            Methanobrevibacter millerae
```

Mapping reads

- The reads are in tiny pieces in the ***.fastq.gz*** file (e.g. 150 base pairs)
- Need to get the computer to do the jigsaw and put the pieces together

1. Without a guiding picture on the box

- **"De Novo Assembly"**
  - Very challenging, time-consuming, error-prone
  - Avoid unless you have to
  - Ideally be working with:
    - a short genome (e.g. bacterial)
    - that is haploid
    - Use a technology with long reads (e.g. *Oxford Nanopore*) for the basic structure ('scaffolds') and then one with extremely accurate reads (e.g. *Illumina Short Reads*) for fixing any small errors ('polishing').

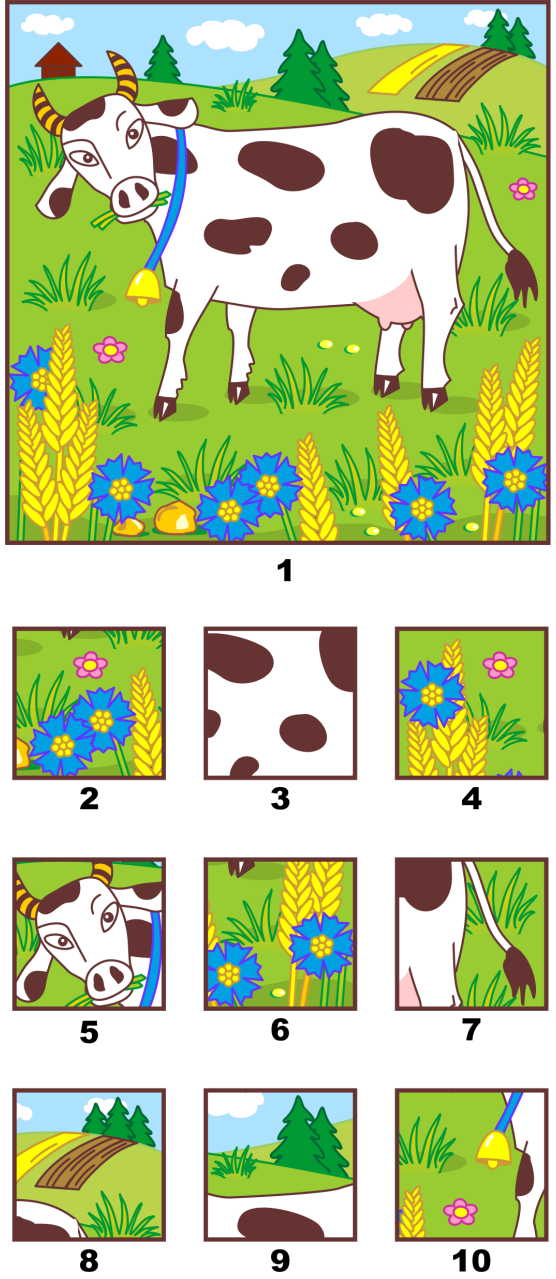


Mapping reads

2. With a guiding picture on the box

- **“Mapping”**
  - Nice and quick,
  - Need to provide a single ‘reference genome’ box picture that you, or someone else, has assembled
    (not the same as imputation ‘reference’!!!)
  - The cow reference genome is *ARS-UCD 1.2*
    - Was assembled by experts over many years
    - Comes from a particular cow in America that was sequenced extremely deeply with a variety of technologies then assembled

Answer: 2,4,7,8



Which pieces don’t map perfectly?

Mapping

- ***BWA*** or ***minimap2*** on NeSI do the mapping, can take a few hours per sample
  - Works out where in the reference genome each read sits
  - Records the read as 'unmapped' if it doesn't fit
    - Fine to have reads from the wrong species, they'll remain unmapped
  - Assigns a quality score to how well the read mapped

S1.fastq.gz

↓

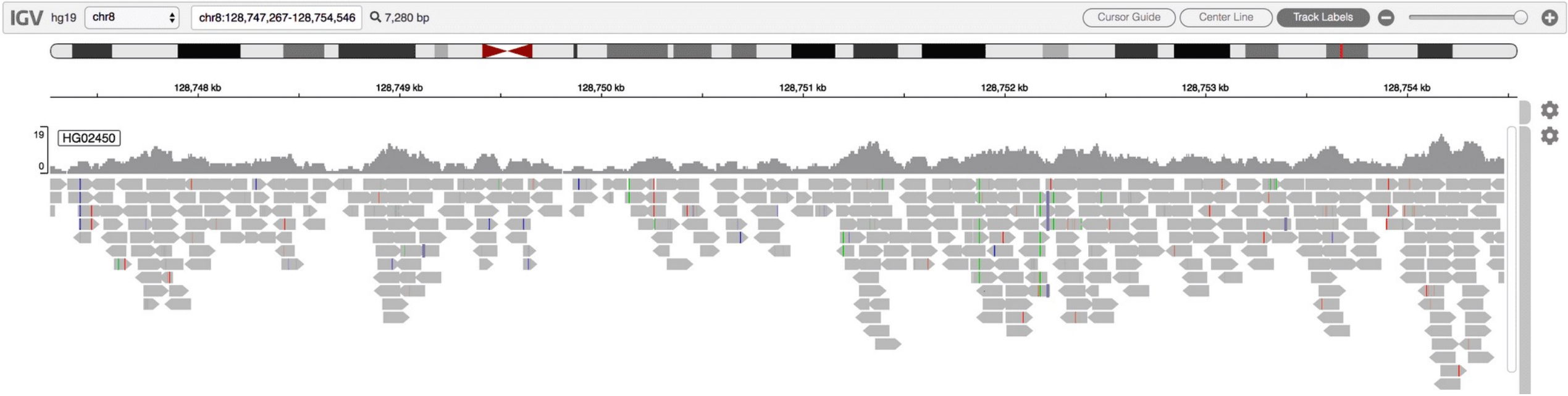
BWA

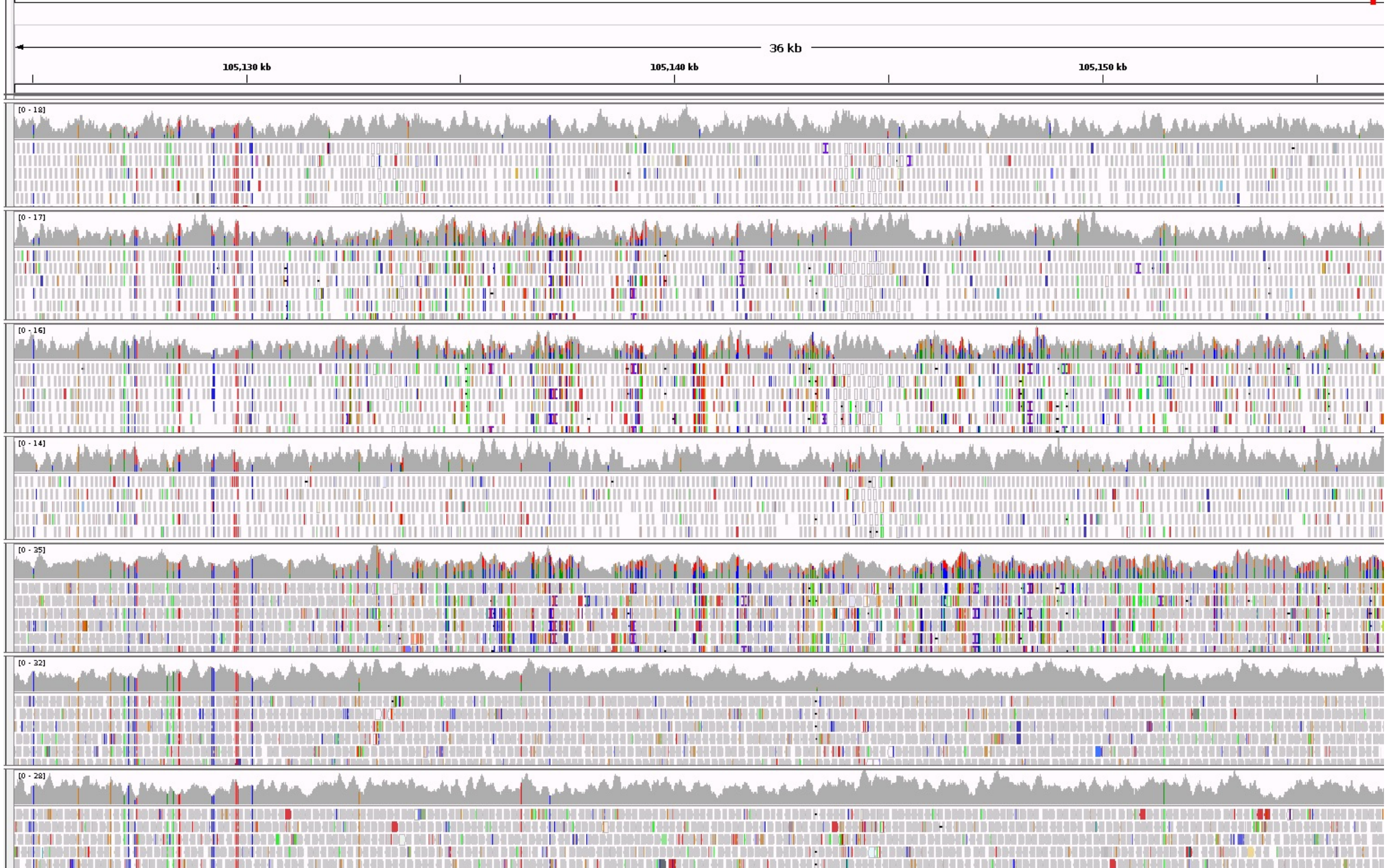
↓

S1.cram

Mapping

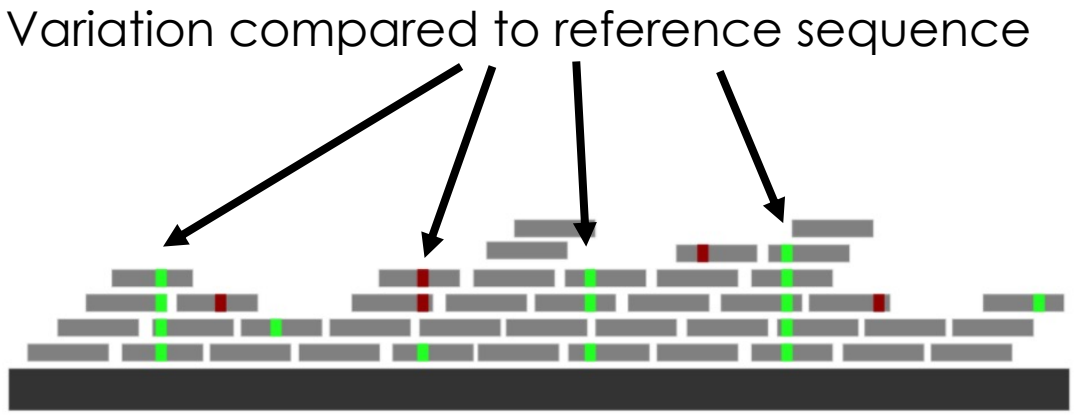
- Produces a **.cram** file (or older/less compressed formats **.bam** or **.sam**)
  - Human readable-ish via: *samtools view <file> | less*
  - Better viewed in ***IGV*** (Integrated Genomics Viewer):




36 kb
105,130 kb
105,140 kb
105,150 kb
[0 - 18]
[0 - 17]
[0 - 16]
[0 - 14]
[0 - 35]
[0 - 32]
[0 - 28]

Variant calling

- Variant caller program looks for places where samples differ to the supplied reference



Analyses the likelihood of:

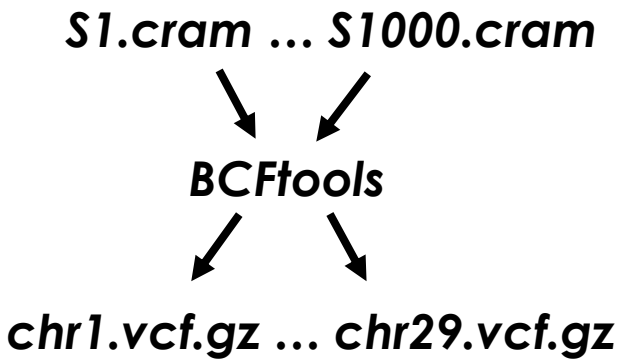
- a sequencing error in the reads

versus

- there being a true difference

Variant calling on NeSI

- ***BCFtools*** or ***GATK*** on NeSI to call variants
  - Calling is done a chromosome at a time, for all samples simultaneously
    - Consider an array job (1 per chromosome), can take a day per chromosome



- The ***.vcf.gz*** format is useful and human-readable-ish (zless <file>.vcf.gz)

```
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO    FORMAT  BBH6865 BBH7134 BBH7235 BBH7236 BBH7237 BBH7238 BBH7239 BBH7240 BBH7241
chr1    2129    .       G       C       281     .       DP=514;VDB=0.900056;SGB=6.02383;RPB=0.0250502;MQB=1;MQSB=1;BQB=0.00167708;MQ0F=0.712062
chr1    2149    .       G       T       515     .       DP=553;VDB=0.216813;SGB=15.3734;RPB=9.82545e-10;MQB=1;MQSB=1;BQB=0.000578554;MQ0F=0.690
chr1    3448    .       ATGACTGATCATGCACTGATCACGTGACT    ATGACT  26.1206 .       INDEL;IDV=2;IMF=0.4;DP=314;VDB=9.27199e-07;SGB=0.404846;MQSB=0.
```

- Fully describes how each sample differs to the reference at each location

Imputation

- Why do imputation in this pipeline?
  - Some sites may have had no reads land on them (so no data)
  - Some sites might have only a few reads land on them (so very poor accuracy of variant calls)
  - Instead of spending much more $$$ to get many more reads from the sequencer, can do imputation
- Feed the variant-called ***chr1.vcf.gz … chr29.vcf.gz*** files into ***Beagle*** to impute
  - Resource intensive!

Two types of Beagle imputation

1. The fast one

- For people working with genotypes panels.
- Beagle's **GT** algorithm is for filling in missing genotypes
  - If you are doing ***genotyping*** rather than ***sequencing***, this is the one you want
  - Nice and quick.
  - Varies with the version:
    - **Beagle 5.0** was a huge improvement on Beagle 4.1.
    - Regressed a bit with 5.1 initially.
    - I haven't tested later versions of 5.1, or 5.2.

S1	0	1	2	1	**?**	1	
S2	0	**?**	2	1	2	1	
S3	1	1	**?**	2	2	**?**	
S4	**?**	1	0	2	0	1	…

Two types of Beagle imputation

2. The slow one

- For people working with *sequence* data.
- Beagle has a **GL** algorithm for reanalysing the variant calls.
  - If you are doing ***sequencing***, you need to run this before you then do the **GT** algorithm, or else your accuracy will be terrible.
  - This is because variant calls from sequencing are *inaccurate*. Beagle needs to do a complex reanalysis of the variant calls taking the haplotypes into account.
  - This is **slow and resource intensive**. ~50 years cpu time for ~7000 cow samples.
  - This is available in **Beagle 4.1** but was not implemented in later versions.

"Recent versions of Beagle do not infer genotypes from genotype likelihood input data, but Beagle versions 4.0 and 4.1 have this capability." – Beagle 5.2 manual

"The GL imputation feature was removed from Beagle 5.0 because version 5 uses a different (much better) phasing algorithm for GT data, and the new phasing algorithm does not naturally extend to handling GL input data. GL imputation is a very hard methodological problem and I think it would be challenging to get funding from my funders to develop methods in this area." – Brian Browning, author of Beagle, email correspondence

Keeping computational resources feasible

- Can easily run out of memory / cpu time with big datasets
- If you have issues with resources
  - Make sure you are only imputing a chromosome at a time
  - Reduce the size of the reference
    - A bigger reference isn't better unless it truly represents the population better
    - Quality of reference much more important than quantity
  - Impute a smaller number of samples at a time
  - As a last resort, split the chromosomes into pieces, impute, and reassemble
    - Will result in loss of accuracy!

A Typical Pipeline

S1.fastq.gz…

Quality check: FastQC → S1.html…

Species identification: Kraken2 → S1.krak…

Mapping: BWA

S1.cram…

Variant calling: BCFtools

chr1.vcf.gz … chr29.vcf.gz

Subsetted into small batches to keep imputation resources low: BCFtools

chr1.aa.vcf.gz … chr1.zz.vcf.gz …

Imputing GL: Beagle 4.1

chr1.aa.GLd.vcf.gz … chr1.zz.GLd.vcf.gz …

Imputing GT: Beagle 5.0

chr1.aa.GTd.vcf.gz … chr1.zz.GTd.vcf.gz …

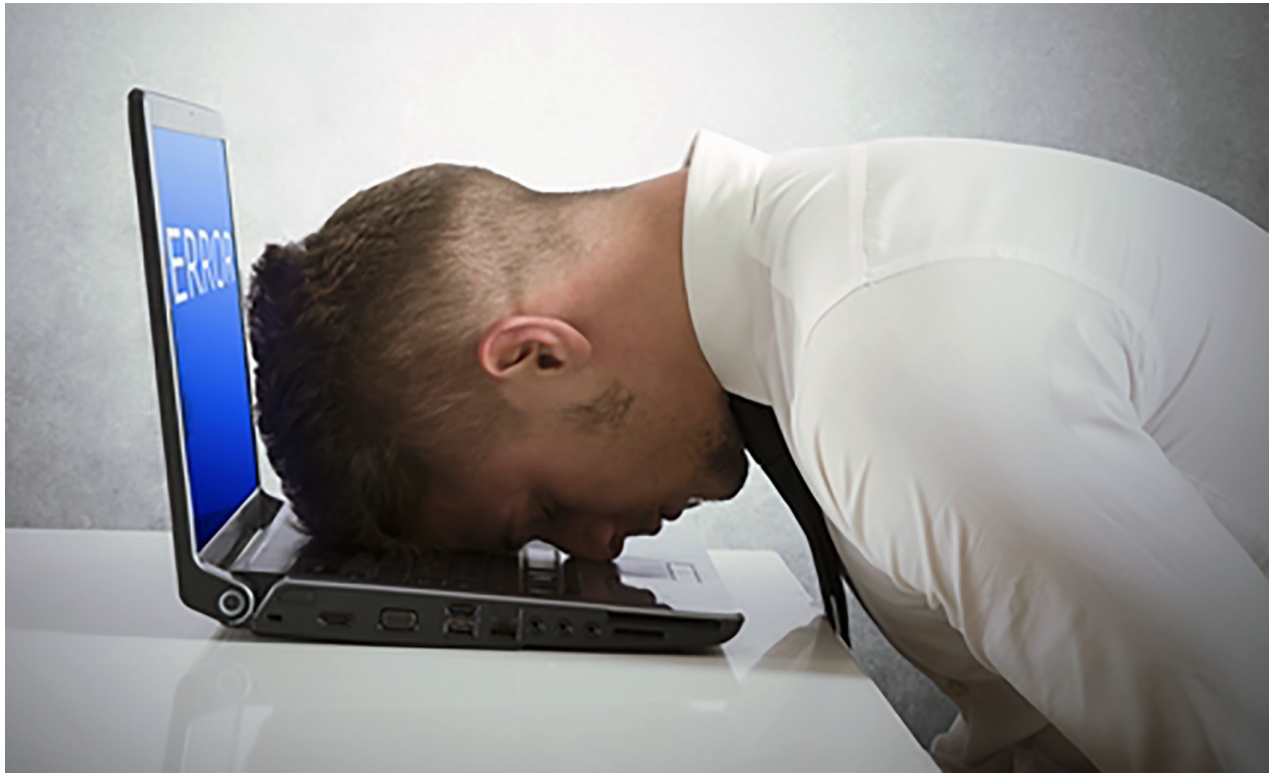
Remerge and unsplit samples: BCFtools

S1.vcf.gz…

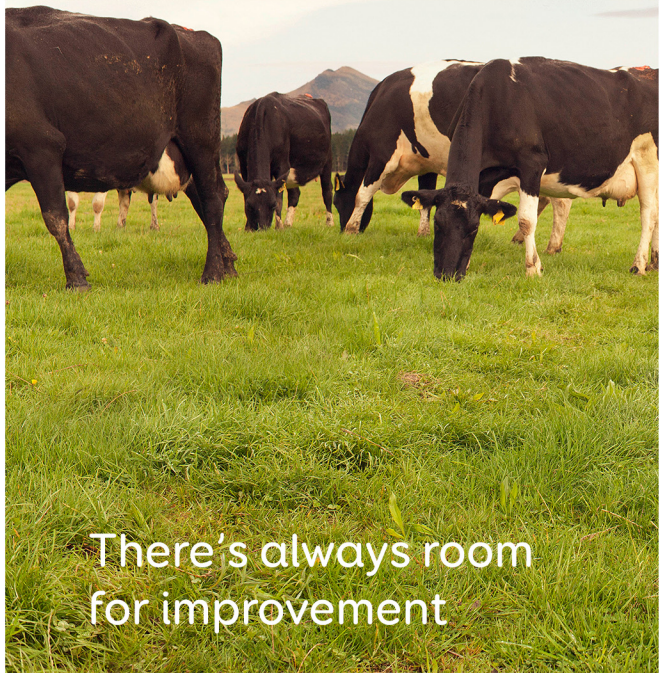
Full genome* of each sample
99% accurate*

User beware

- Each individual program in these pipelines
  - Is not particularly user friendly
  - Has its own special quirks, and settings
  - Manuals can be hard going
  - Needs a lot of testing around resource usage
  - Can be hard to get right, time consuming, and frustrating



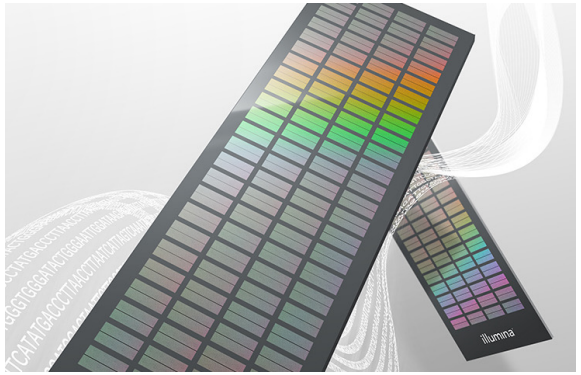
LIC DNA Sequencing Service



- LIC accepts customer prepared libraries for sequencing
- Our QC ensures optimal cluster generation and maximal data output for each run
- A full range of NovaSeq flowcells are available to suit each project
- Contact liam.williams@lic.co.nz
  027 403 1801

LIC®

Summary



Imputing GT: Beagle 5.0

Identification: Kraken2

Mapping: BWA

Variant calling: BCFtools

Imputing GL: Beagle 4.1

Imputing GT: Beagle 5.0

Questions?