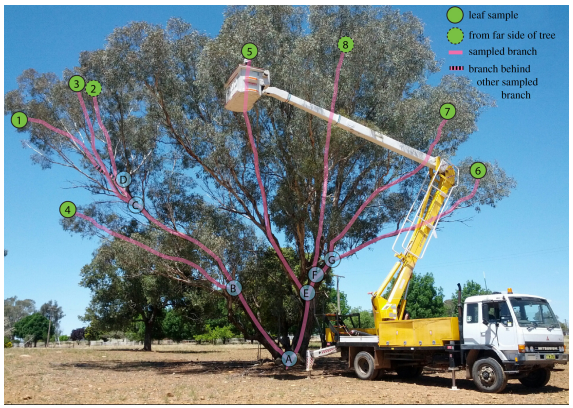
Methods for Detecting Mutations in Non-model Organisms

Adam Orr

11/6/20



Why Care About Mutations and Genotyping?

Human Health

- Cancer
- Personalized Medicine

Agriculture

- Looking for interesting phenotypes in clonally reproducing species
- Breeding programs

Evolution

- Mutations are the ultimate source of variation
- Mutation rate diversity and evolution

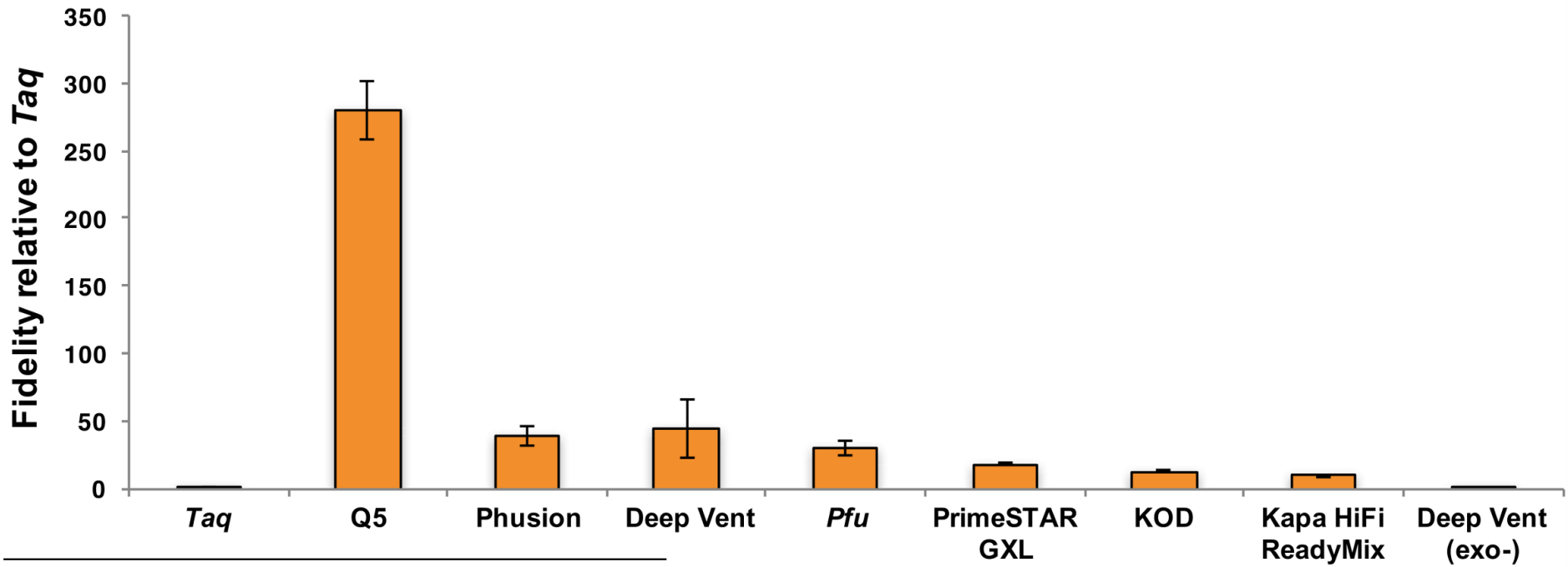
[0] https://commons.wikimedia.org/wiki/File:White_nectarine_and_cross_section02_edit.jpg

Mutations can be difficult to detect

Mutations are very rare, but sequencing errors are very common.

Sequencing error alone is $\sim 10^{-3}$ while mutation rate after error-checking is $\sim 10^{-9}$

- Errors accumulate during PCR prior to sequencing - then propagate.
- *Taq* $\sim 10^{-4}$
- Technical error from sequencer



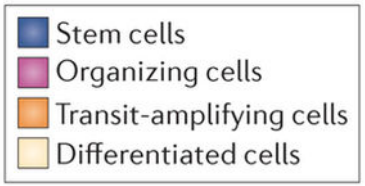
[0] Potapov V, Ong JL (2017) Examining Sources of Error in PCR by Single-Molecule Sequencing

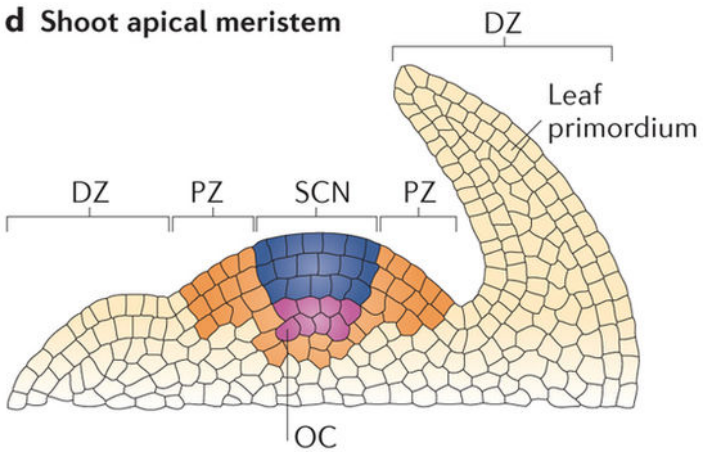
Working with Non-model Organisms can be difficult

- No reference genome
- Many methods assume a reliable reference and other supporting information
- Assembling your own is possible but unsatisfying; costly and time-consuming to do well
- 50,000 species in NCBI genome database of 600,000 in taxonomy database; few are reference quality
- We need robust reference-free methods!

How does plant growth affect somatic mutation rate?

We want to understand mutation patterns within a non-model organism.

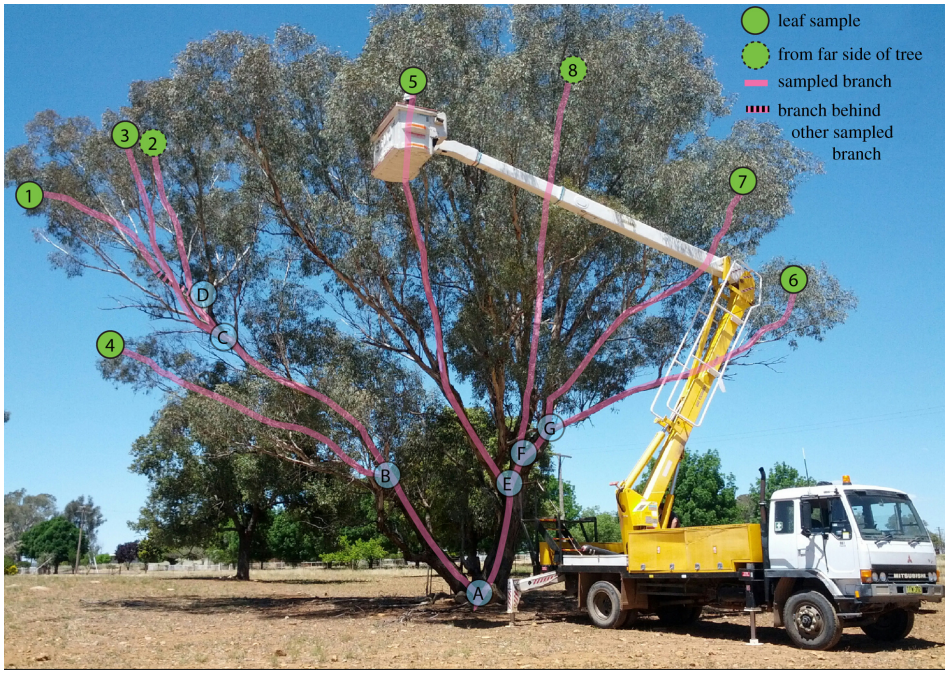




- The genetic structure of the plant *should* mirror its physical structure.

[0]Heidstra & Sabatini (2014) Plant and animal stem cells: similar yet different. doi:10.1038/nrm3790

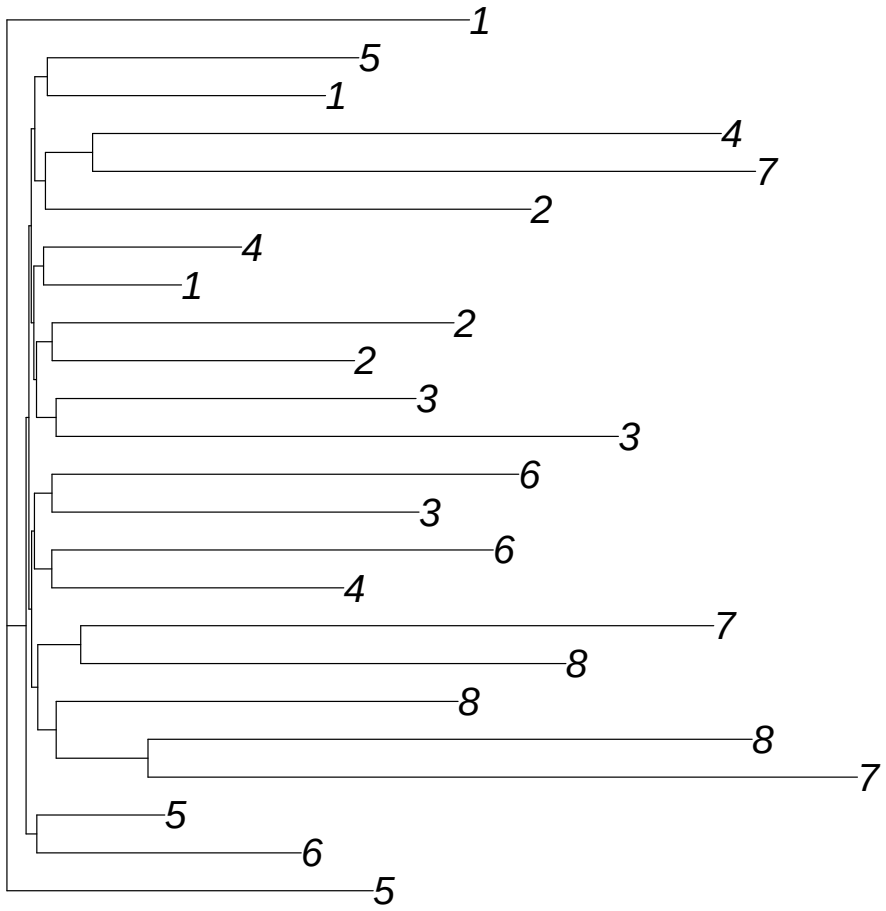
A Genetic Mosaic



- Mosaic: differential oil production gives protection from beetles
- Does the pattern of mutation match the physical structure?
- Can we detect enough mutations to measure the mutation rate?

[0] Orr et al. (2020) A phylogenomic approach reveals a low somatic mutation rate in a long-lived plant. doi:10.1098/rspb.2019.2364

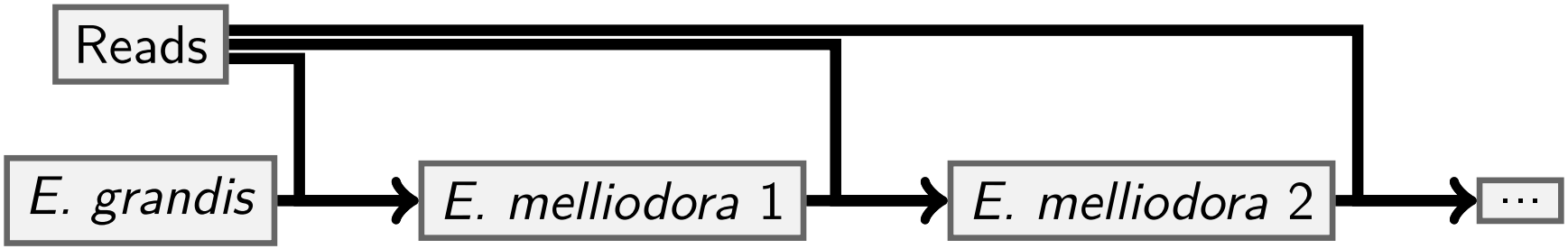
Current reference-free methods are insufficient



- Sequence 8 samples in triplicate
- $\sim$10X coverage for each replicate
- DiscoSNP++ uses small differences in similar sequencing reads to find potential mutations
- Coverage may not be sufficient for this method
- The repetitive nature of the genome may make it difficult to differentiate repeated DNA from mutations

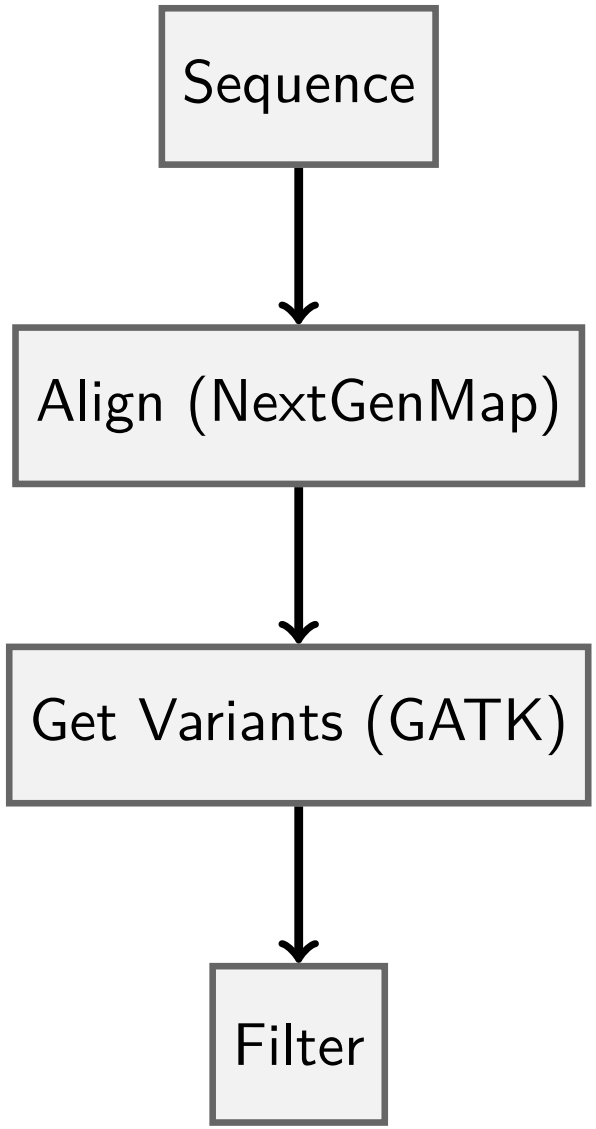
Approximating a Genome

Use *E. grandis* genome as a starting place, then generate a new reference and map to that reference.

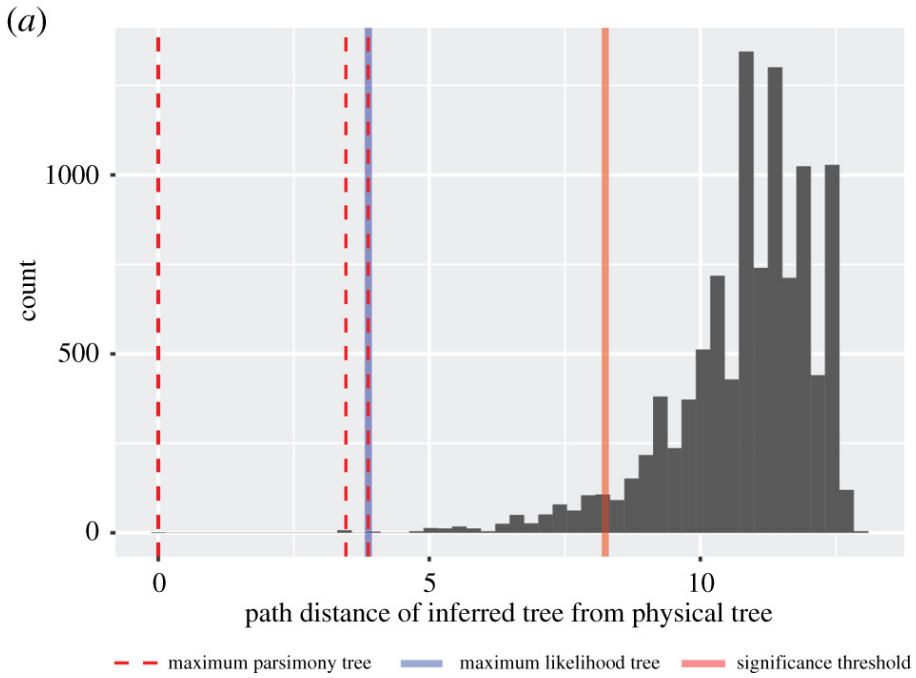


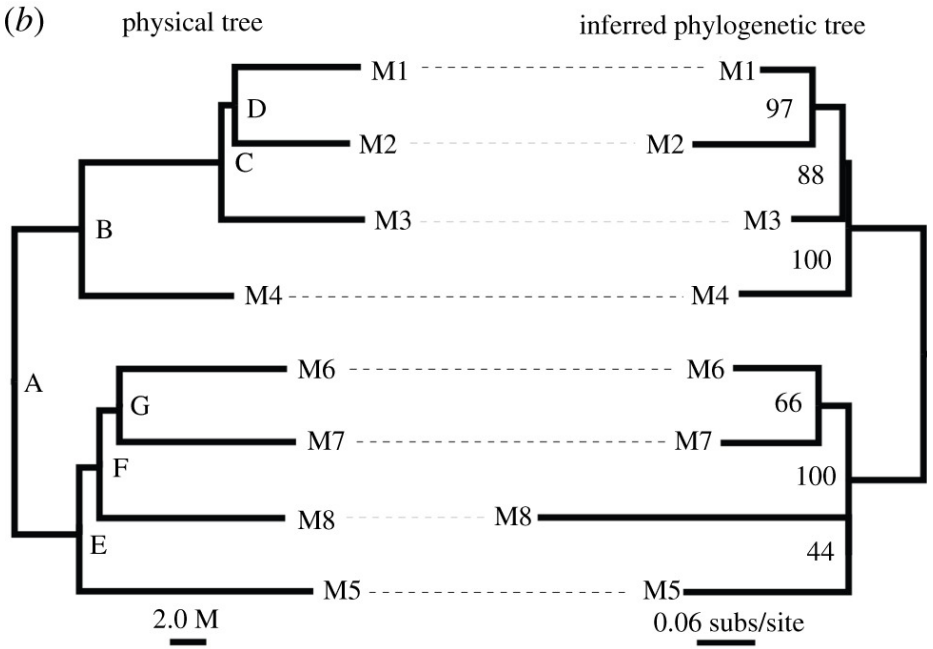
Analysis Pipeline

- Sequence 8 samples in triplicate
- Align sequence to the edited *Eucalyptus grandis* genome
- Use replicates to remove false positives



Pipeline Produces Tree Close to Physical Tree





[0]Orr et al. (2020) A phylogenomic approach reveals a low somatic mutation rate in a long-lived plant. doi:10.1098/rspb.2019.2364

Using Tree Topology Gives Higher Recall Rate

- Thus, it's reasonable to assume the physical topology when inferring mutations
- *DeNovoGear* is a variant-calling method that uses information in the tree topology to call variants.
- By simulation, we introduced 14000 mutations on the tree

GATK	*DeNovoGear*
3859 mutations	4193 mutations
27%	30%

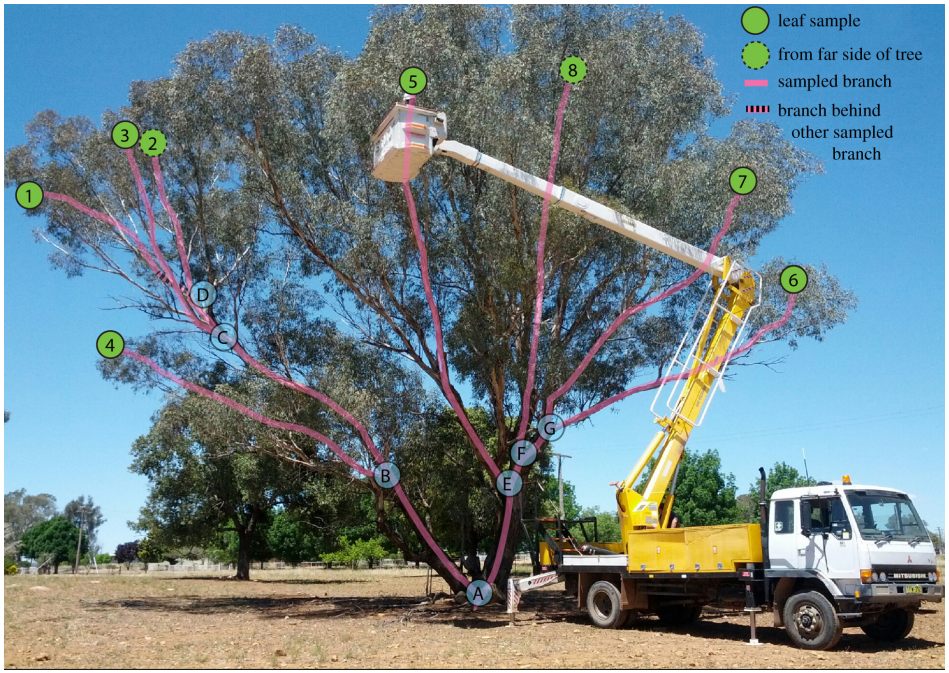
Using Random Trees to Estimate False Positive Rate

- If we assume mutations should match the tree structure, no real mutations should also match a random maximally-distant tree.
- Simulate 100 trees maximally distant from the true tree and ask how many the pipeline detects on average.

GATK	*DeNovoGear*
55.71 of 99 mutations	.11 of 90 mutations
56.3%	.12%

Mutation Rates

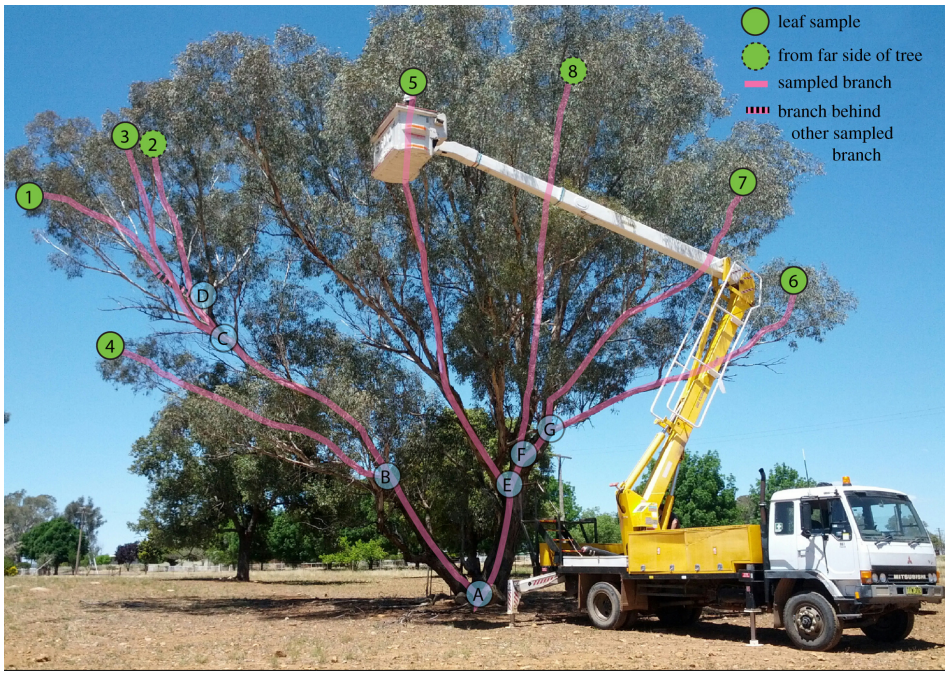
- Detected 90 mutations.
- 20 mutations in genes.
- Estimated recall of $\sim 30\%$.
- $90 \times \frac{1}{.3} = 300$ mutations.
- ~ 3.3 mutations per meter of length
- 2.7×10^{-9} mutations per base per meter
- Somatic mutations account for ~ 55 mutations per leaf tip.



Population Estimates

We studied *one* individual, but we can make conjectures about the population.

- The average height of a eucalypt is 22.5 M
- Mutation rate per base, per generation from somatic mutation is 6.2×10^{-8}
- We estimated $\theta = 0.025$
- Since $\theta = 4N_e\mu$, $N_e = 102,000$



This per-generation rate is $\sim 10\times$ larger than *Arabidopsis*, but *Eucalyptus* is $100\times$ larger.

How do we do better? Base Quality Scores help find errors

Errors make variant calling difficult - but we can predict them.

- FASTQ format data has a quality score
- Quality scores represent $P(error)$ on a phred scale.

$$P(error) = 10^{\frac{-Q}{10}}$$

$$Q = -10 \log_{10} P(error)$$

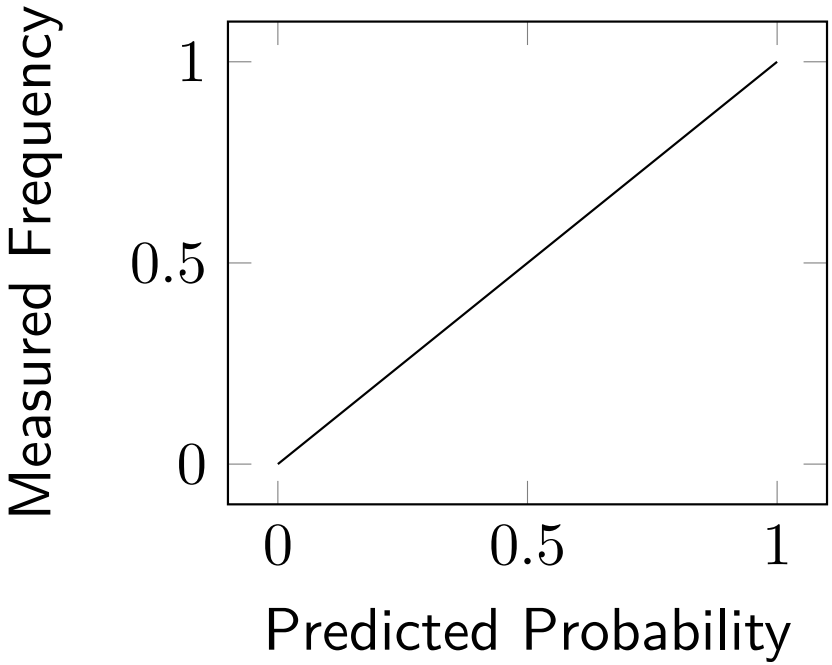
FASTQ Example

```
@SEQ_ID
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
+
!''*((((***+))%%%++)(%%%%).1***-+*''))**55CCF>>>>>>CCCCCCC65
```

[0] https://en.wikipedia.org/wiki/FASTQ_format

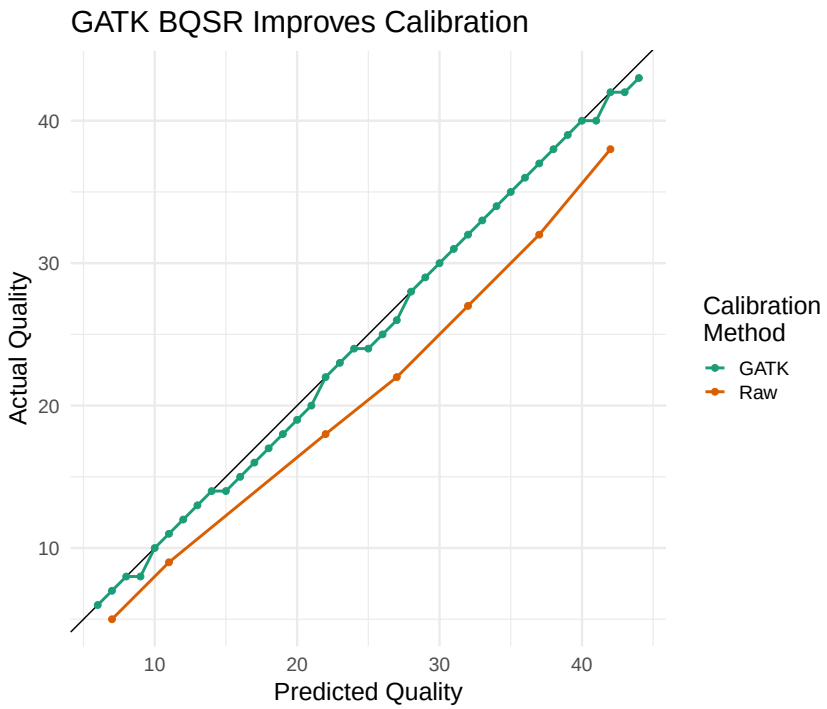
Quality scores are predictions

- A quality score is a **prediction** about whether a base call is correct.
- Predictions are said to be **calibrated** if the predicted event occurs as often as predicted.
- The weather forecast contains a **prediction** about whether it will rain.
- If it rains on a day with a 30% chance of rain, what does that mean?



Quality scores aren't well-calibrated

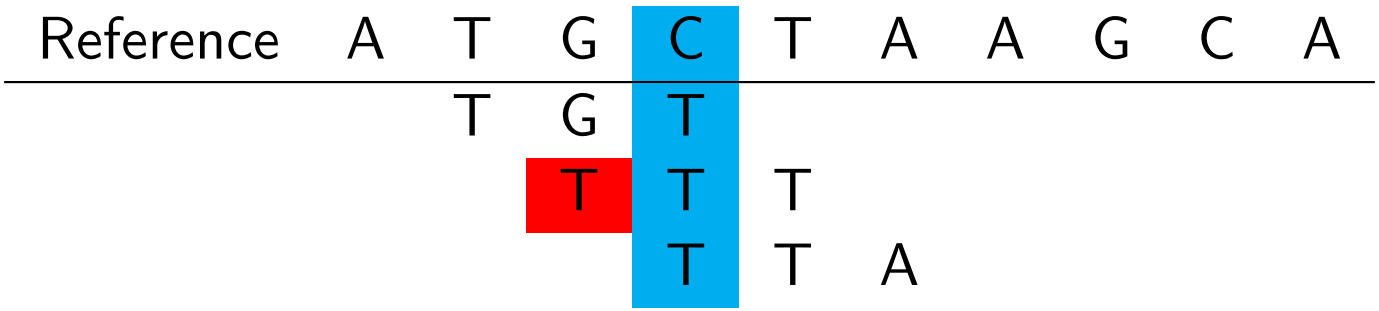
- If quality scores *were* well-calibrated, it would be easier to identify errors
- Base Quality Score Recalibration can be done to fix calibration issues.
- GATK method for BQSR require a database of variable sites in your data then assumes mismatches at nonvariable sites are errors.



Base Quality Score Recalibration

GATK BQSR is the standard method for BQSR. It works in 3 phases:

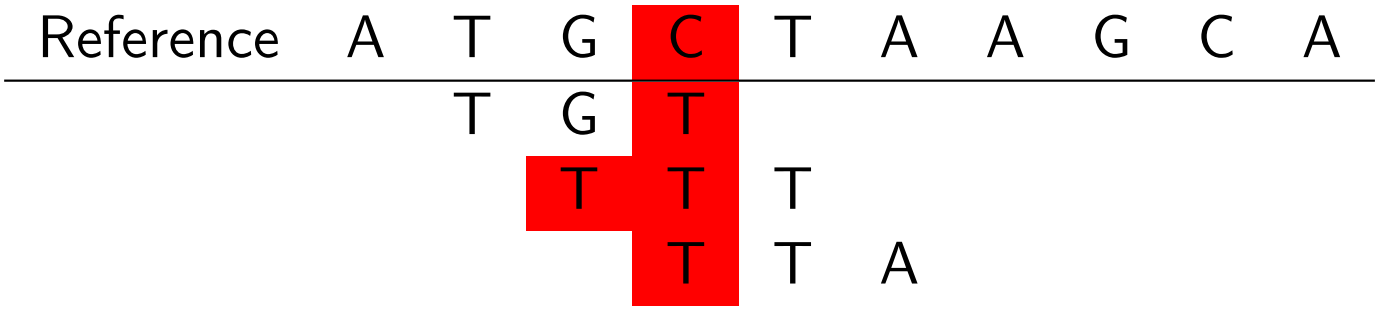
1. Find errors with alignment
2. Train model
3. Recalibrate with model



One site is excluded and one base is an error. $\frac{1}{6}$ bases is an error.

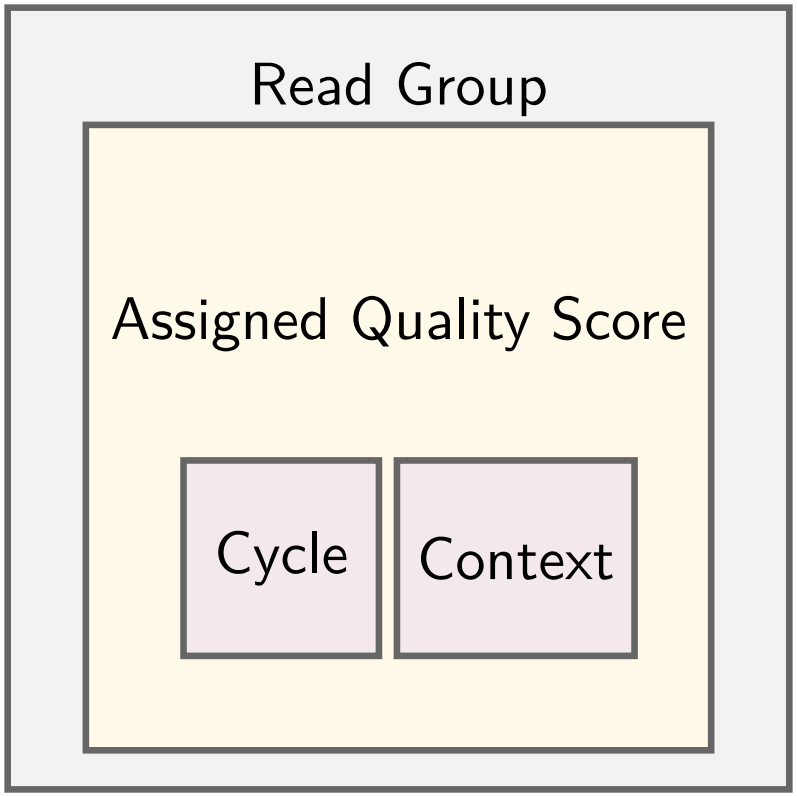
GATK BQSR is difficult in non-model organisms

- Many mismatches between reference and sample (if there is one)
- No database with sites to exclude

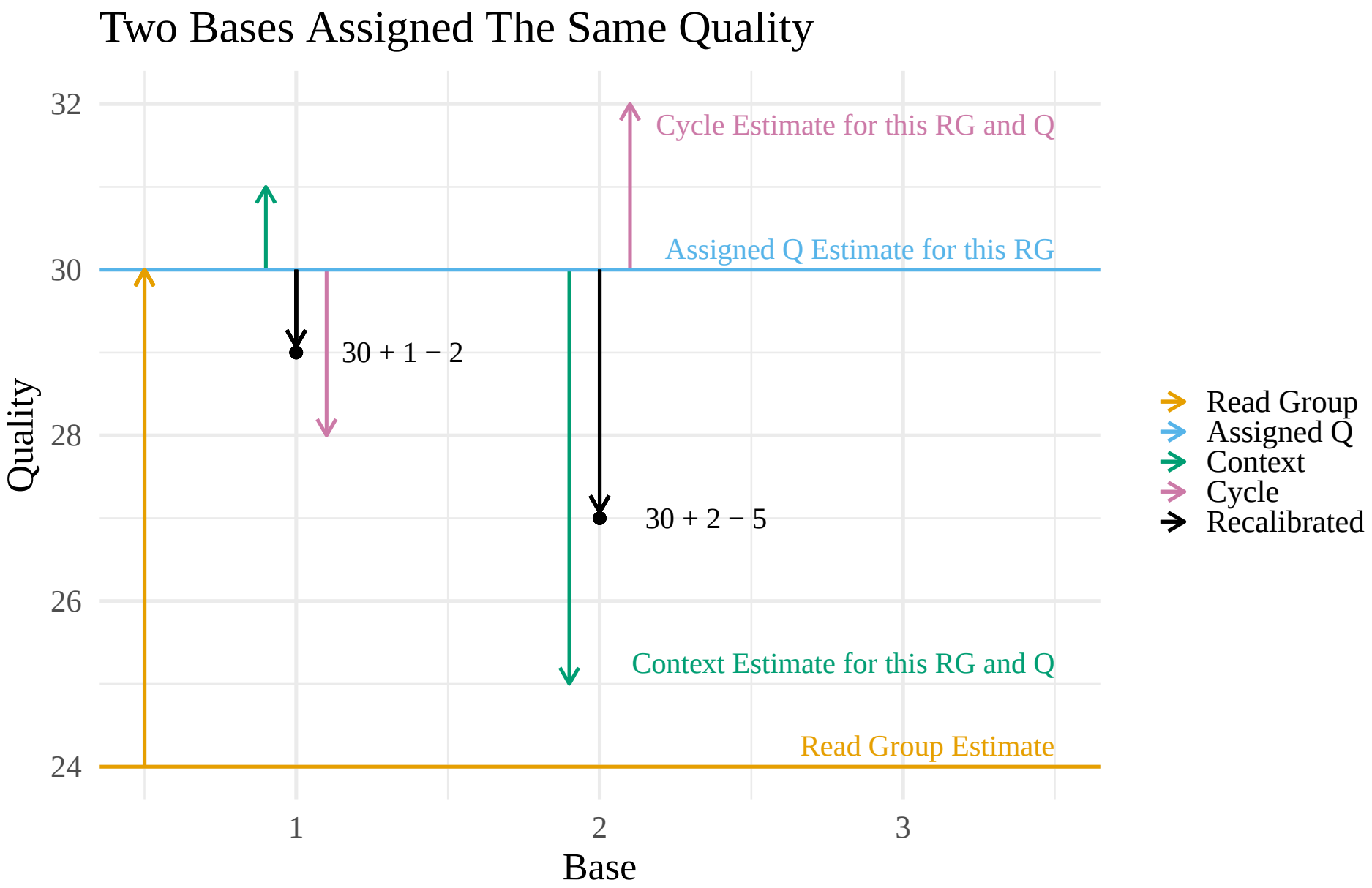


One site isn't excluded; only one base is really an error. $\frac{4}{9}$ bases are estimated to be errors.

GATK BQSR uses a hierarchical linear model to determine how much to adjust each quality score



Example Recalibration



GATK BQSR uses a hierarchical linear model to determine how much to adjust each quality score

$$Q = \bar{Q} + \Delta RG + \Delta Q + \Delta C(Cycle) + \Delta X(Context)$$

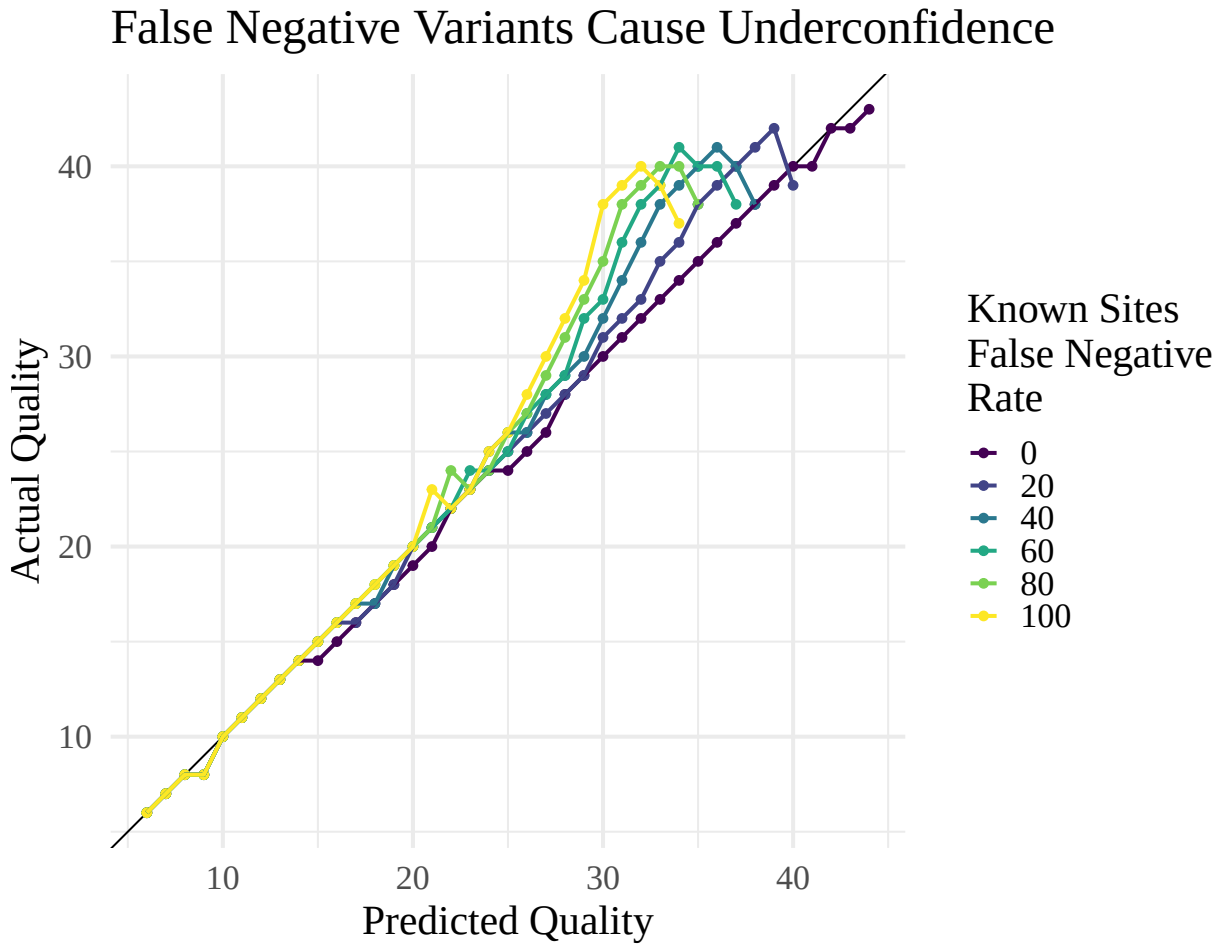
$$\Delta RG = argmax_q\{P(\mathcal{B}(RG_t, q) = RG_e) * P(\mathcal{N}(\bar{Q}) = q)\} - \bar{Q}$$

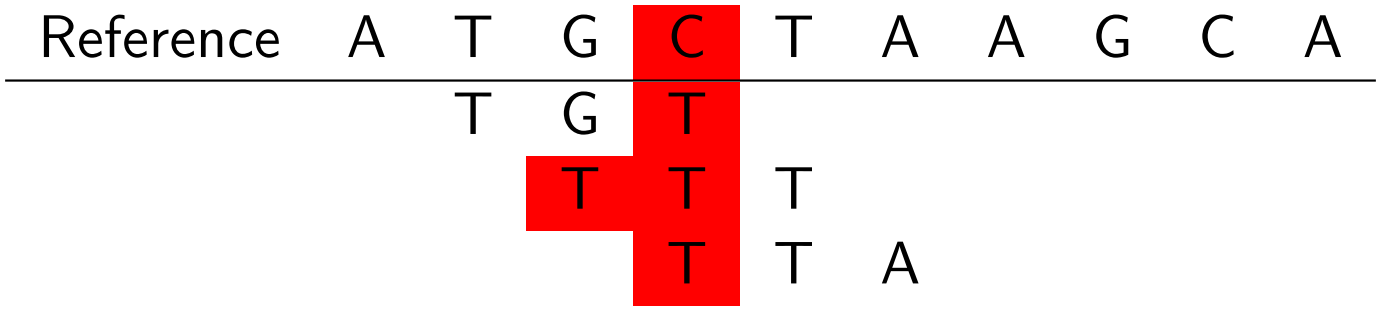
$$\Delta Q = argmax_q\{P(\mathcal{B}(Q_t, q) = Q_e) * P(\mathcal{N}(\bar{Q} + \Delta RG) = q)\} - (\bar{Q} + \Delta RG)$$

$$\Delta Cycle = argmax_q\{P(\mathcal{B}(C_t, q) = C_e) * P(\mathcal{N}(\bar{Q} + \Delta RG + \Delta Q) = q)\} - (\bar{Q} + \Delta RG + \Delta Q) \quad (1)$$

$$\Delta Context = argmax_q\{P(\mathcal{B}(X_t, q) = X_e) * P(\mathcal{N}(\bar{Q} + \Delta RG + \Delta Q) = q)\} - (\bar{Q} + \Delta RG + \Delta Q) \quad (2)$$

BQSR vulnerable to simulated false negatives





Alternative approaches get around using a database of variable sites

- SOAP2 has a consensus calling model that performs BQSR
- ReQON limits the number of errors there can be at a site
- Synthetic spike-ins
- GATK Recommended method: Use the best reference you can get, call a confident set of variants, and use that.

Let's find errors with k-mers instead of an alignment

- Error correction methods exist that use k-mers to identify errors rather than an alignment and reference.
- Most error correctors don't update quality scores; `Lighter` optionally updates quality scores of corrections to a value but this doesn't materially affect the calibration.

Error Detection with Lighter

1. Subsample k-mers at rate α
2. Use subsampled k-mers to find trusted k-mers
3. Use trusted k-mers to correct untrusted k-mers

Read	A	T	G	C	T	A	A	G	C	A
Kmer 1	A	T	G							
Kmer 2		T	G	C						
Kmer 3			G	C	T					

Subsampling

Read	A	T	G	C	T	A	A	G	C	A
Kmer 1	A	T	G							
Kmer 2		T	G	C						
Kmer 3			G	C	T					

Keep each k-mer with probability α. If the same sequence is sequenced M times, that k-mer will appear multiple times. The probability we sample it is then $1 - (1 - \alpha)^M$

Copy 1	A	T	G
Copy 2	A	T	G
Copy 3	A	T	G

P(not sampled) $= (1 - \alpha)^3$ so P(sampled) $= 1 - (1 - \alpha)^3$

Trusting K-mers

If we **assume an error will only show up at most twice**, the probability an erroneous k-mer is sampled is $1 - (1 - \alpha)^2$

Each base pair has between 1 and k associated k-mers; we can do a binomial test to determine whether the number of sampled k-mers associated with a base pair is too high for it to be an error:

$$P(\mathcal{B}(covering, 1 - (1 - \alpha)^2) = sampled) > .95$$

When there are K base-pairs that are trusted, we add this to a set of trusted k-mers.

Finding Errors in reads

Given the set of trusted k-mers, find the longest stretch of trusted k-mers in the read. Then, the bases that border this stretch are errors.

Read	A	T	G	C	T	A	A	G	C	A
Kmer 1	A	T	G							
Kmer 2		T	G	C						
Kmer 3			G	C	T					
Kmer 4				C	T	A				

CTA isn't trusted, try CTC, CTG, CTT. Maximize the number of trusted k-mers in the read this way.

Implementation Improvements

A **Bloom filter** stores sampled and trusted k-mers

- Hash the kmer and use bits from the hash to set bits in the filter; to test membership, see if those same bits are set.
- You should choose the size of the bloom filter based on the number of entries and a desired false positive rate.

- Lighter estimates the number of entries to be $1.5\times$ the genome length, but this is too low.
- Lighter uses a patterned bloom filter, but doesn't use aligned blocks of memory.

Estimated Number of Sampled Kmers

$$K_{\text{total}} = \sum_{\text{Reads}} \text{Read length} - k + 1 \tag{3}$$

$$= (\text{Read length} - k + 1) \times \text{Number of reads} \tag{4}$$

$$< \text{Read length} \times \text{Number of reads} \tag{5}$$

$$< \text{Read length} \times \frac{\text{Coverage} \times \text{Genome length}}{\text{Read length}} \tag{6}$$

$$< \text{Coverage} \times \text{Genome length} \tag{7}$$

So the expected number of sampled k-mers is

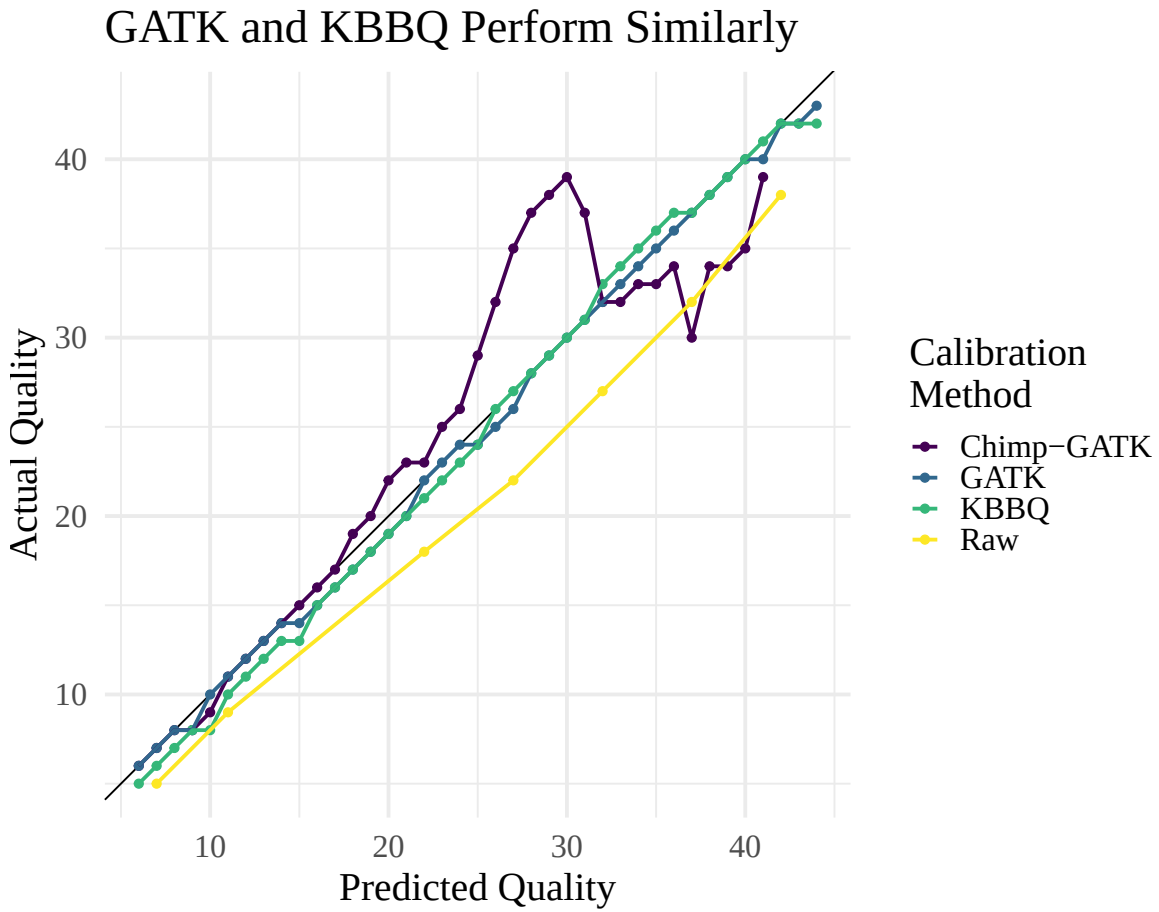
$$E[K_{\text{sampled}}] = E[\mathcal{B}(x; \alpha, K_{\text{total}})] \tag{8}$$

$$= \alpha \times K_{\text{total}} \tag{9}$$

$$< \alpha \times \text{Coverage} \times \text{Genome length} \quad < 7 \times \text{Genome length} \tag{10}$$

K-mer Based Base Quality score recalibration works

- Combining error correction and BQSR is effective
- Method implemented in `kbbq` software



Methods that rely on accurate quality scores suffer

BCFTools' multiallelic caller estimates allele frequency for a site as:

$$f_a^s = \frac{\sum_{b=a} Q_a}{\sum_b Q_b} \tag{11}$$

$$f_a = \frac{\sum_s^S f_a^s}{S} \tag{12}$$

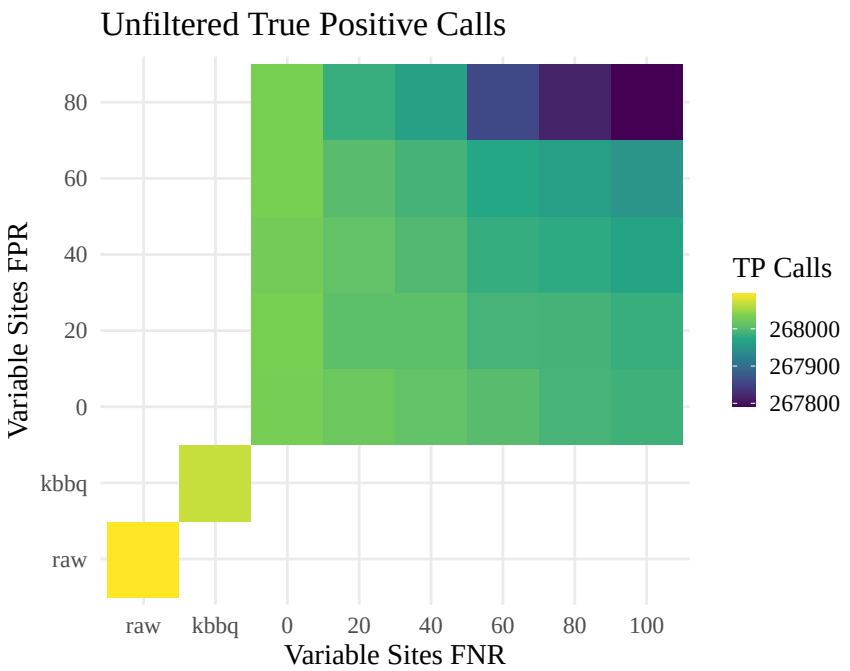
Suppose a systematic effect reduces the quality score for allele T. While the true allele frequency of A is .5, we might get data like: 5 A (Q 40), 5 T (Q 30) for every sample.

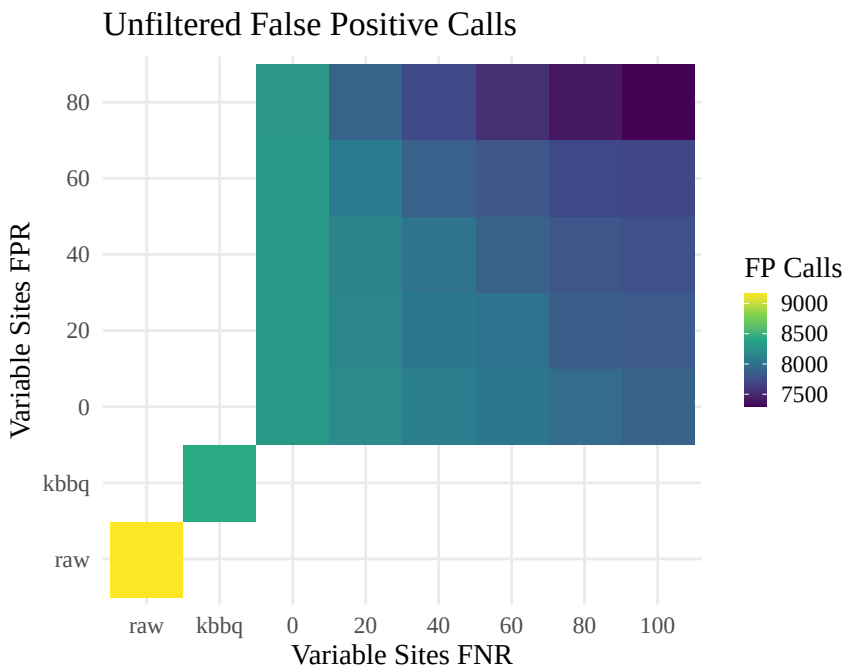
The calculated allele frequency is $\frac{40*5}{40*5+30*5} = .57$

Does BQSR help?

- Some find BQSR improves minor allele detection, especially in high coverage data
- Heng Li found that replacing quality scores with the lower of the quality score and the mapping alignment quality (MAQ) improved accuracy of heterozygote detection
- BQSR is time-consuming
- BQSR takes extra hard-drive space

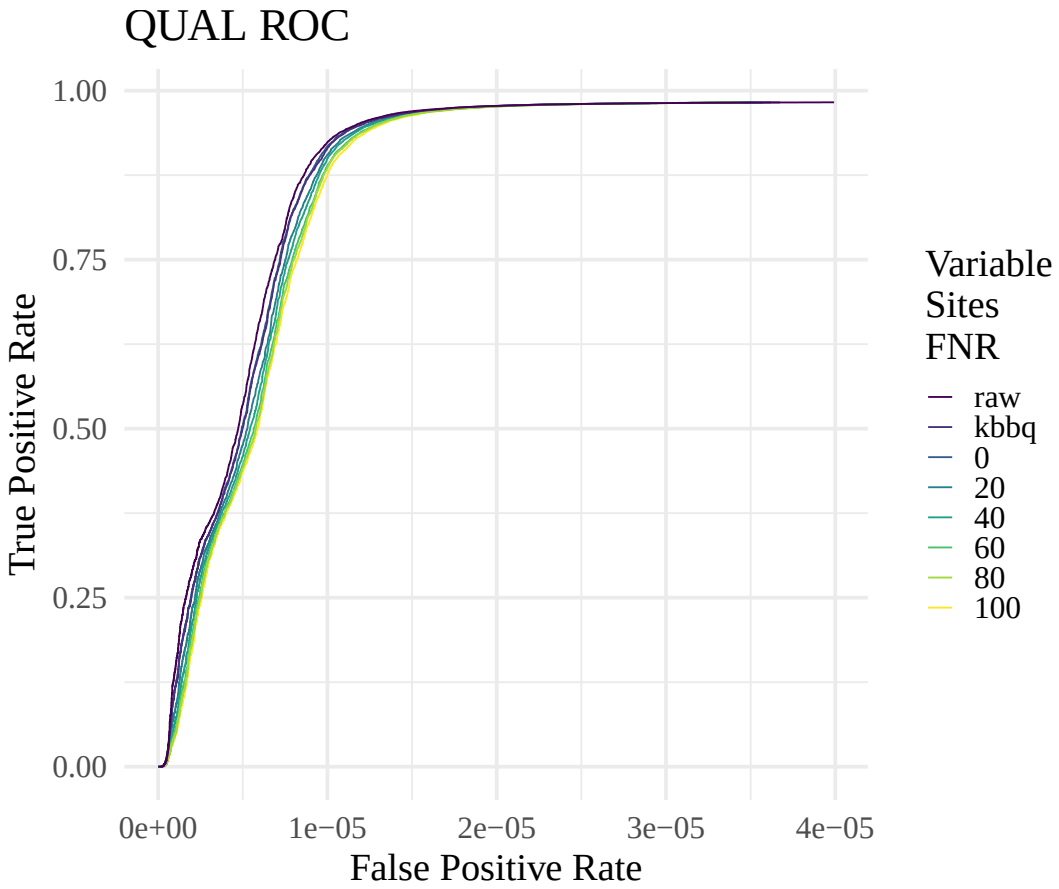
Improved Calibration Increases Number of Positives - But Not More Than Raw Scores



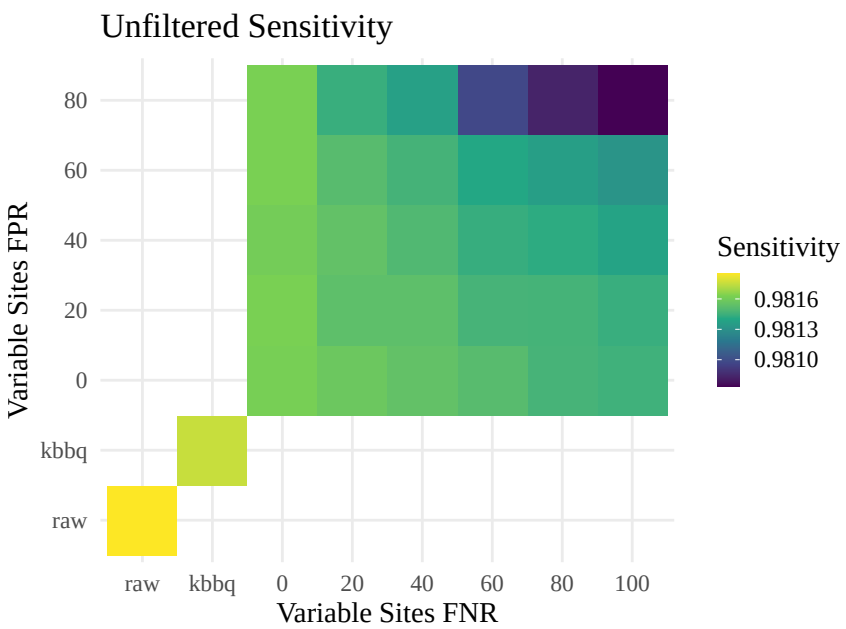


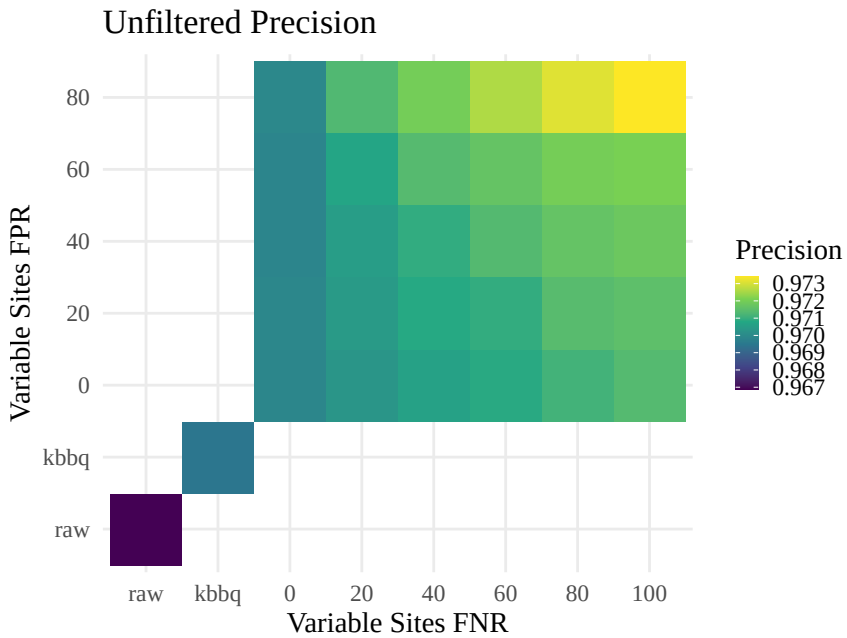
Calibration Changes Variant QUAL Annotation

The QUAL field represents the likelihood a site is variable.



The Difference in Sensitivity and Precision is Small





Despite Benchmark Data Results, Recalibration Improved *E. melliodora* Calls

	Raw	GATK	KBBQ
Num Variants	88	99	106
Estimated False Positives	36.54	55.71	35.54
Previously-identified Positives	34	30	34
Estimated New True Positives	18	13	36

What could explain this discrepancy?

- Well-developed protocols for DNA extraction for human cell-lines
- Base callers are tuned to work well for human data
- Less variety in quality scores may be good for calling

Acknowledgements

- My committee
- My lab: Dr. Cartwright
  - Abby
  - Juan
  - Ziqi
  - Courtney
  - Aleks

KBBQ: `https://github.com/adamjorr/kbbq`
Dissertation: `https://github.com/adamjorr/dissertation`

This work was supported by grants from the NIH, NSF, BSF, and the Graduate College Completion Fellowship