avocado: A Variant Caller, Distributed

Frank Austin Nothaft, Peter Jin, Brielin Brown
{fnothaft, phj, brielin}@berkeley.edu

Background

Three stages in modern DNA processing pipelines:

1. **Sequencing:** Generate 100-250 base pair reads
2. **Alignment:** Align these reads to the reference genome
3. **Variant Calling:** Determine gene variants & genotypes

Variant calling is an interesting area: "Accurate" algorithms are slow and don't scale (60 hrs/genome), and are inaccurate for high complexity regions (error is $> 75\%$).

Goals:

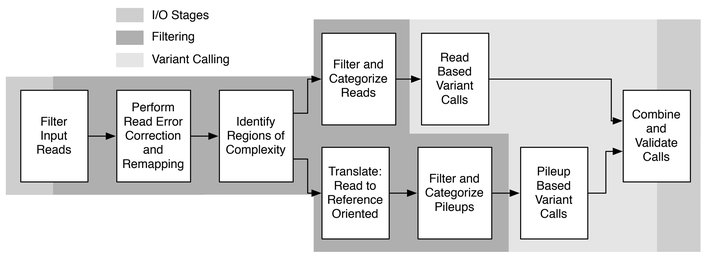
1. Build a variant caller designed for distributed computing
2. Develop an open-source alternative to the GATK

Pipeline

Tech Specs:

- Built in Scala on top of Parquet and BDAS Spark
- Leverages new ADAM read/pileup/variant call format
- Scalability well past 30+ nodes; other pipelines are limited to 26 (1/chromosome)

Pipeline:



Design Principles:

- Reuse read processing stages from ADAM
- Use mapping quality/coverage as filtering heuristic
- Design is modular: easy to add new calling algorithms

Performance

Local Assembly

Base SNP Calling

For calling SNPs on a single sample, we look at genome loci that show evidence of a SNP (at least one non-reference base). Genotype likelihoods are calculated by:

$$\mathcal{L}(g) = \frac{1}{m^k} \prod_{j=1}^{l} (m-g)\epsilon + g(1-\epsilon) \prod_{j=l+1}^{k} (m-g)(1-\epsilon) + g\epsilon$$

m = ploidy, g = genotype state, ϵ = likelihood of error,
l = bases matching reference, k = bases at locus

Genotyping is biased towards the reference. We compensate by the allele frequency and call a non-reference genotype if $g \in (1, 2)$ has the highest probability.

Sufficient Statistics/Joint Calling