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:

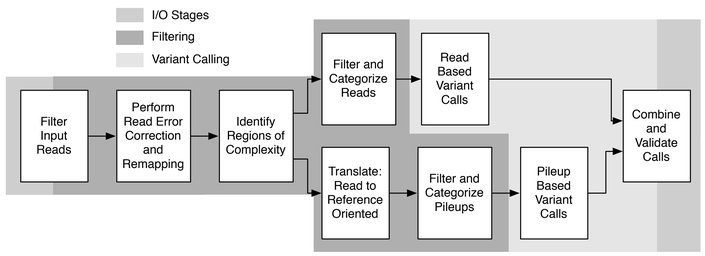
1. Build a variant caller designed for distributed computing
2. Develop an open-source alternative to the GATK

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:



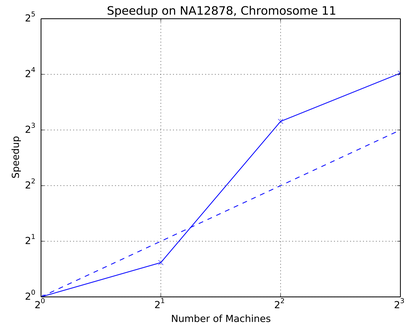
Design Principles:

- Use mapping quality/coverage as filtering heuristic
- Use assembly methods on high complexity regions
- Design is modular: easy to add new calling algorithms

Performance

Notes:

- Algorithm is currently disk bound due to shuffles: performance bug in pileup creation due to partitioning
- Plan to fix performance bug by doing interval-based rod conversion:
  - Lump reads by reference position group to maintain locality
  - Fewer objects created than reads → pileups → rods



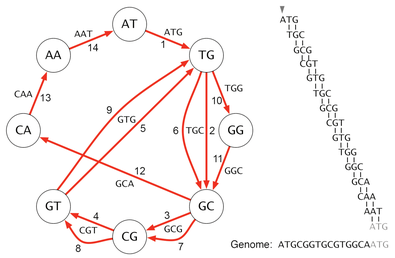
MapQ,Cov	0.2	0.4	0.6
40	88%	54%	20%
50	90%	56%	22%
60	98%	91%	88%

% Reads in High Complexity Region

Sample	Time	Size	Machines
Venter	6hr	99 GB	20× cr1.8xlarge
NA12878, chr11	0.47 Hr	673 MB	4× m2.4xlarge
NA12878, chr20	0.25 Hr	297 MB	4× m2.4xlarge

Performance Over Different Datasets

Local Assembly



We partition the high-complexity locations into regions and use local k-mer assembly to discover the most likely haplotype pair per region. The likelihood of a pair of haplotypes H_j and $H_{j'}$ is given by:

$$\mathcal{L}(H_j, H_{j'}) = \prod_i \left[\frac{P(r_i|H_j)}{2} + \frac{P(r_i|H_{j'})}{2} \right]$$

where r_i is a read. We obtain $P(r|H)$ by a pairwise HMM alignment model.

Figure credit: P.E.C. Compeau, P.A. Pevner, G. Tesler, "How to apply de Bruijn graphs to genome assembly," Nature Biotech. 29(11), 2011.

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

For a few samples, one may look-up the MAF ϕ in a reference and compensate the the single sample likelihood

$$\hat{g} = \arg\max_g \mathcal{L}(g)\mathbf{P}(g|\phi)$$

When many samples are collected it can be desirable to compute a population MAF while performing genotype calling. For each SNP a, this is done via EM:

$$\phi_{a,t+1} = \frac{1}{M} \sum_{i=1}^{N} \frac{\sum_{g_i} g_i \mathcal{L}(g_i)\mathbf{P}(g_i|\phi_{a,t})}{\sum_{g_i} \mathcal{L}(g)\mathbf{P}(g|\phi_{a,t})}$$

$M = \sum_i m_i$ = total number of chromosomes N = number of individuals