An algebra of single nucleotide variants

Johannes Köster, Sven Rahmann

Genome Informatics, University of Duisburg-Essen

Introduction

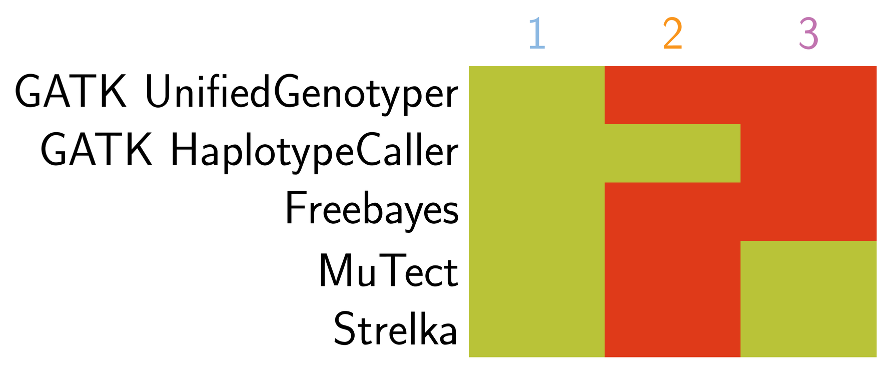
Variant calling on NGS data often entails filtering samples against each other to e.g.

- detect de-novo mutations (child vs. parents, tumor vs. normal),
- eliminate sequencing artifacts.

This gives rise to three problems.

1. Insufficient evidence problem: The filtering fails if the coverage is too low.
2. N + 1 problem: Calling samples in groups helps with the insufficient evidence problem. But the addition of a sample leads to expensive redundant calculations.
3. FDR problem: The obtained variant qualities do not reflect the filtering. This makes controlling the false discovery rate (FDR) difficult.

Currently, there appears to be no variant caller to solve them all.



We present the ALgebraic PArallel CAller (**ALPACA**), a variant caller that combines a flexible algebraic variant calling approach with preprocessed HDF5-based index data structures and a parallel, OpenCL-based implementation to solve all three problems.

A flexible query language

For a set of samples S, define a query language $\mathcal{Q}$ as the smallest set of formulas with

$$s \in \mathcal{Q}_S$$
$$\phi_1 \oplus \phi_2 \in \mathcal{Q}_S$$
$$\phi_1 \ominus \phi_2 \in \mathcal{Q}_S.$$

This allows all kinds of filtering scenarios to be formulated, e.g.

- Call all variants in a group of samples:
$$s_1 \oplus s_2 \oplus s_3 \oplus \dots$$
- Call somatic mutations in e.g. a tumor sample s_t compared to a healthy blood sample s_b:
$$s_t \ominus s_b$$
- Call de-novo mutations in a metastasis sample compared to a tumor and a healthy blood sample:
$$s_m \ominus (s_t \oplus s_b)$$
- Call somatic mutations in a group of tumors s_t, s_t' compared to their normals s_b, s_b':
$$(s_t \oplus s_t') \ominus (s_b \oplus s_b')$$
- Do the same in a paired way:
$$(s_t \ominus s_b) \oplus (s_t' \ominus s_b')$$

Algebraic variant calling

Let V_s be the set of true (but unknown) variant loci of sample $s \in S$. We strive to approximate a map φ, such that

$$\varphi(s) := V_s$$
$$\varphi(\phi_1 \oplus \phi_2) := \varphi(\phi_1) \cup \varphi(\phi_2)$$
$$\varphi(\phi_1 \ominus \phi_2) := \varphi(\phi_1) \setminus \varphi(\phi_2).$$

For any $\phi \in \mathcal{Q}$, we therefore calculate the posterior probability of observing zero alternative alleles ($M = 0$) subject to ϕ at a given genomic locus i as

$$\Pr(M=0|\phi,i) := \begin{cases} \Pr(M=0|D_{i,S'}) & \text{if } \phi = \bigoplus_{s\in S'\subseteq S} s \\ 1-\Pr(M>0|\phi_1,i)\cdot\Pr(M=0|\phi_2,i) & \text{if } \phi = \phi_1 \ominus \phi_2 \\ \Pr(M=0|\phi_1,i)\cdot\Pr(M=0|\phi_2,i) & \text{otherwise} \end{cases}$$

Then, we estimate the variant loci subject to ϕ as

$$\varphi(\phi) \approx \varphi^*_\alpha(\phi) := \{i \mid \forall i = 1,2,\dots,n : \Pr(M=0|\phi,i) \le \alpha\}.$$

Even low coverage evidence for a variant in a sample used for filtering will affect the resulting posterior.

Controlling FDR

FDR can be controlled to not exceed α^* by setting the threshold

$$\alpha = \max\{\alpha' \in [0,\alpha^*] \mid \overline{FDR}_{\alpha'} \le \alpha^*\}$$

with

$$\overline{FDR}_\alpha = \frac{1}{|\varphi^*_\alpha(\phi)|} \sum_{i \in \varphi^*_\alpha(\phi)} \Pr(M=0|\phi,i).$$

Since the obtained posterior probabilities reflect the given query, controlling the FDR becomes easy.

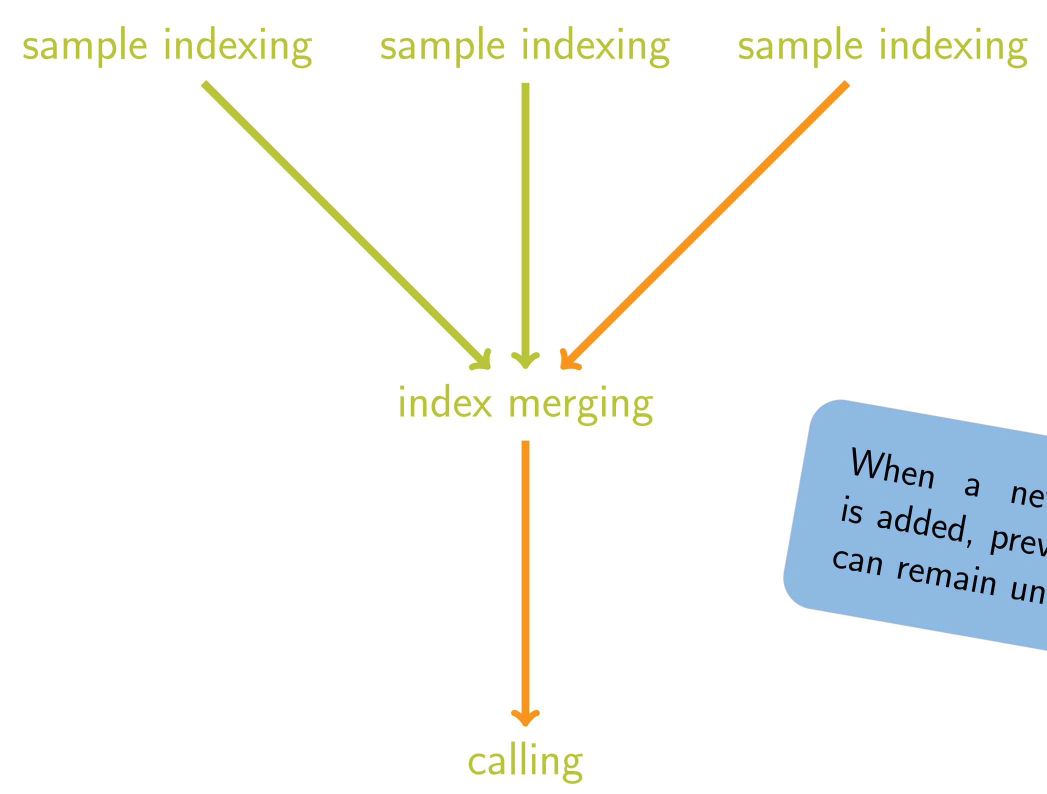
Preprocessing into HDF5 index

The probability $\Pr(M=0|D_{i,S})$ is calculated from per-sample allele frequency likelihoods

$$\Pr(D_{i,s}|M=m)$$

in a bayesian way (similar to GATK; dePristo et al. 2011). $D_{i,s}$ is the pileup of read bases of sample s at locus i. The likelihoods are independent of the query formula ϕ.

- Hence, allele frequency likelihoods for all covered loci can be preprocessed into per-sample HDF5 indexes.
- Sample indexes can be merged into a global index, keeping only loci with a maximum likelihood allele frequency $\neq 0$.
- Calling with different queries becomes a matter of seconds.



When a new sample is added, previous ones can remain untouched.

Fig. 1: ALPACA workflow. Adding a new sample (orange) requires only the repetition of merging and calling.

LaTeX TikZposter