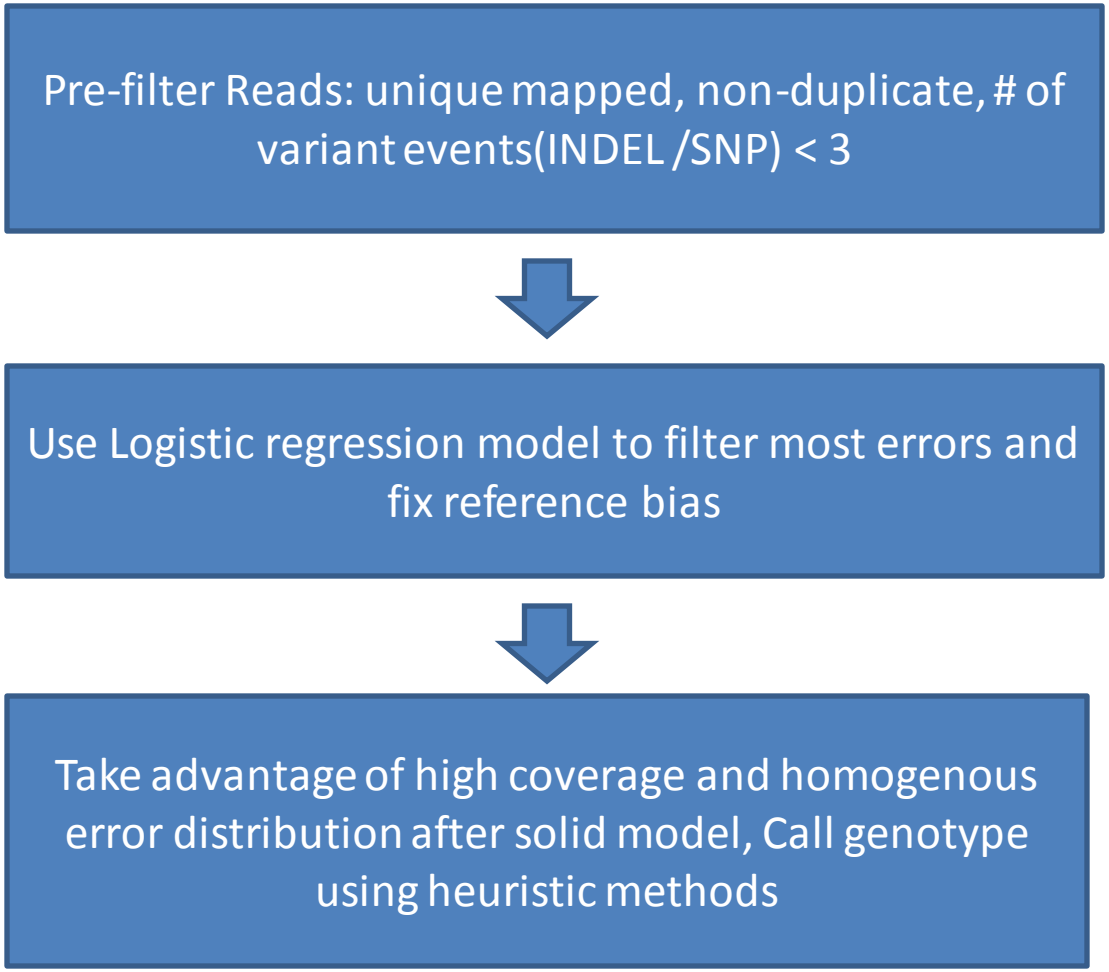
Methods and preliminary results of solid-genotyper

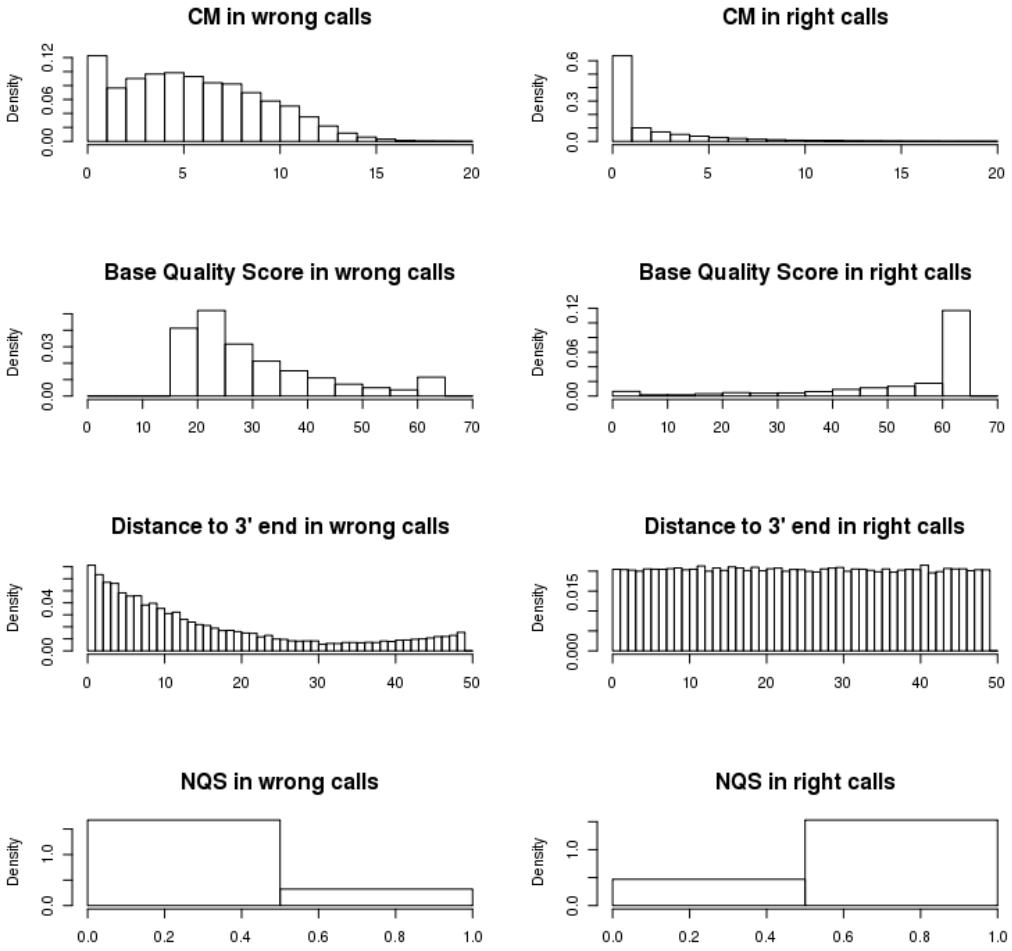
Jin Yu in Fuli's lab

Feb 10th 2011

Workflow of solid-genotyper

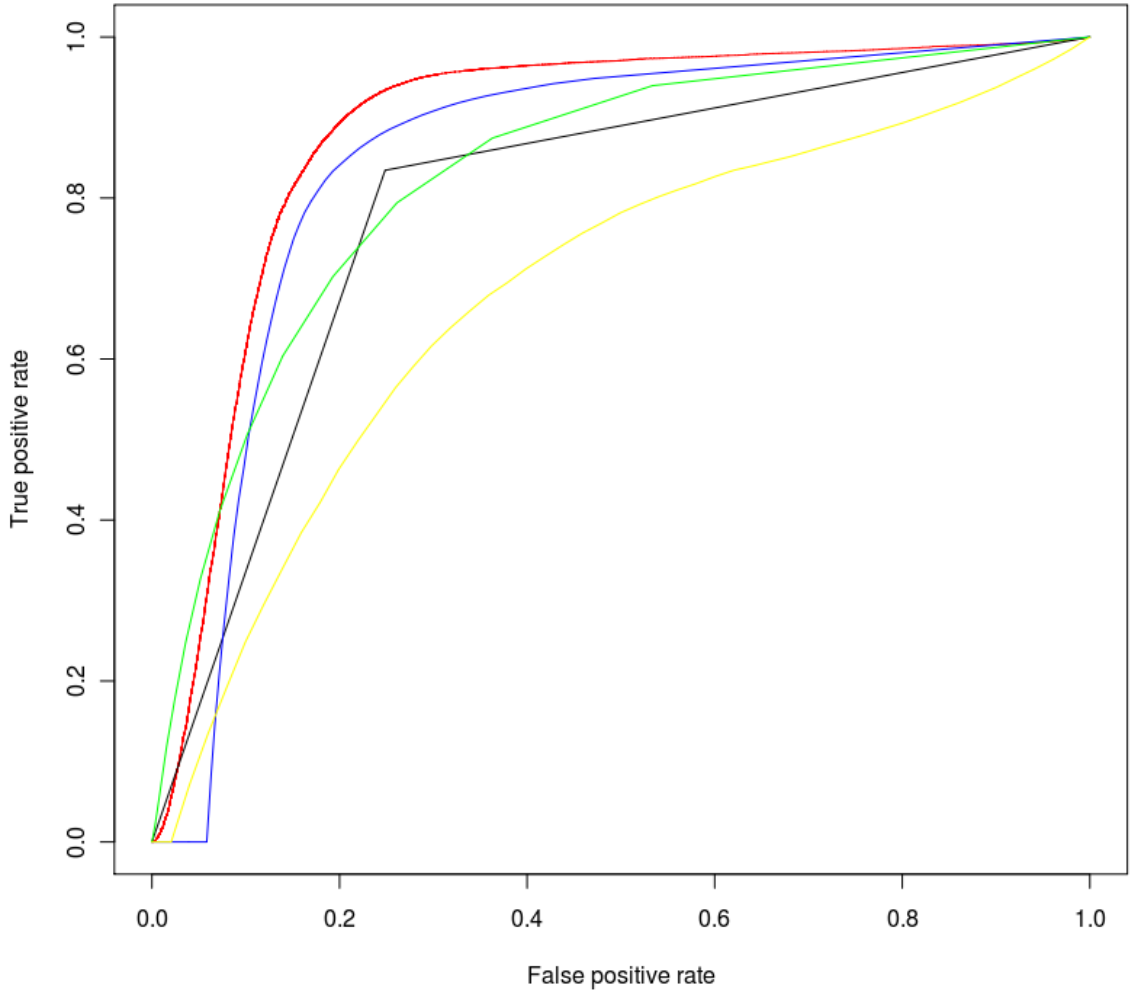


Characterize SOLiD error model



- Methods:
  - Using BFAST to map SOLiD reads to a E.coli strand
  - Known few true variant sites, differences are treated as errors
- Variables used:
  - CM (number of color corrections occurred in this read)
  - Raw base quality score
  - Distance to 3` end
  - NQS (Neighboring Quality Score)

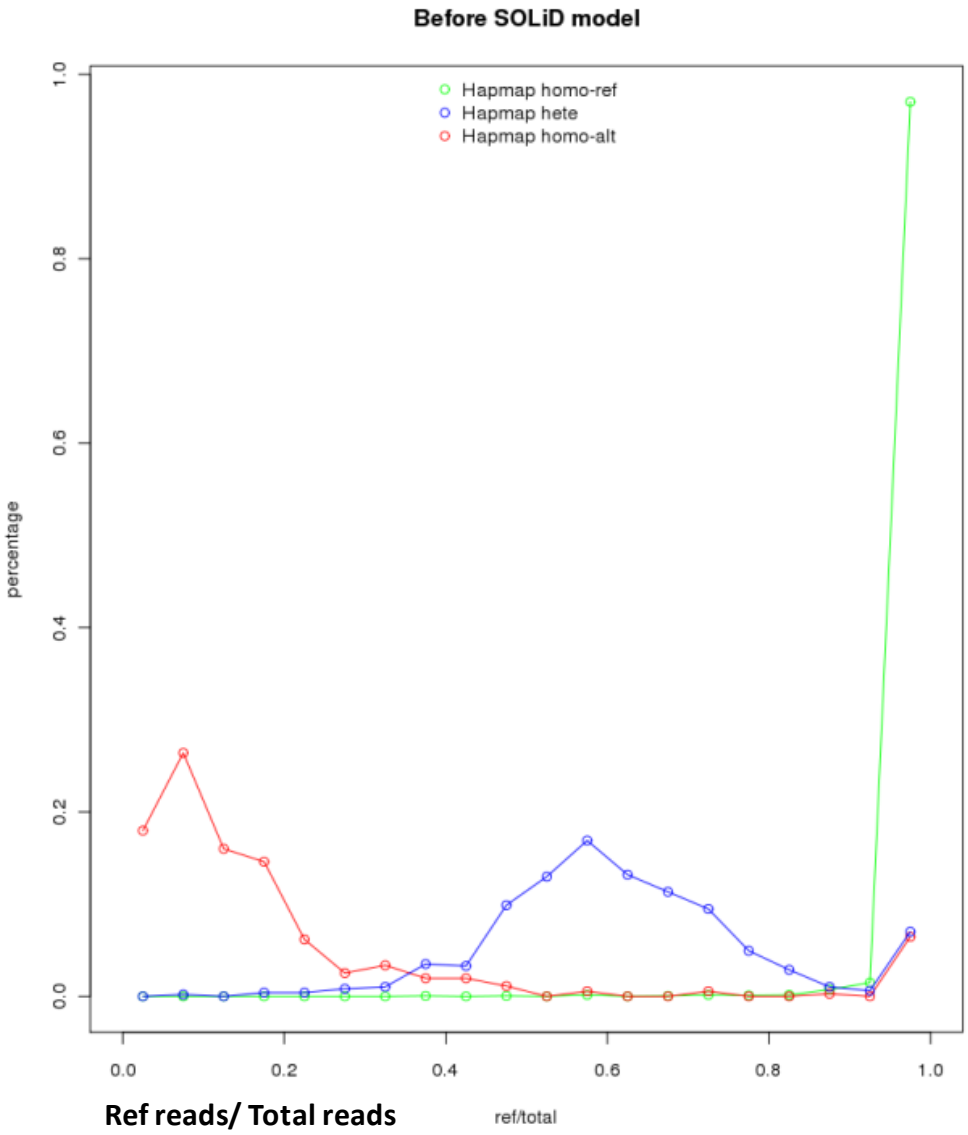
Logistics regression on reads level



Performance summary:

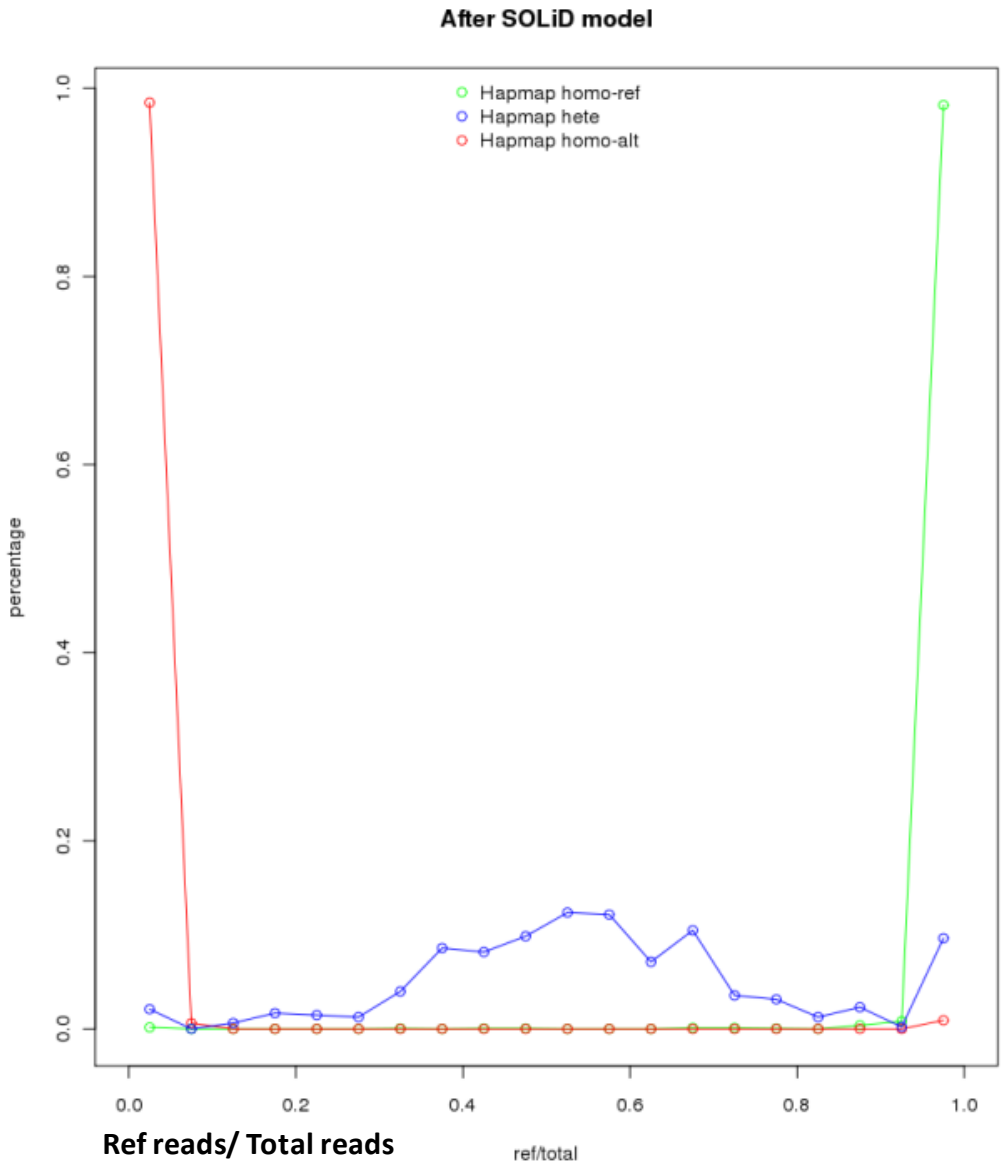
- logit predictor has better performance than any single variables
- Filter ~90% errors at the cost of ~15% coverage depth
- Preferable to mark mapping errors (results shown later)

Reference bias in raw alignments



- Cannot survive even in high coverage (average coverage ~60X in this case)
- Causes:
  - Relative short read length (50bp)
  - Special treatment on SOLiD alignment (provided by BFAST)
  - Solve the color space reads ambiguity in a way to maximum the mappablity
  - always turn ambiguous calls to the reference base

Corrected allele distribution after solid-genotype processing



Fix the reference bias at the cost of ~16% coverage depth

- Turn the contradicted calls from reference back to N, account for <1% (GATK recalibration probably will also do it)
- Turn the base at the end of 3` end to N, account for 2%
- Base calls failed logit model, account for ~14%

Heuristics methods to call genotype

- Minimal total effective reads depth to get a confident call (currently use 8)
- Ratio of total effective read depth to call one allele (currently use 0.1)
- Minimal effective read depth to call one allele (currently use 2)

Implementation

- The prototype was implemented in Ruby
- The production version was implemented in C
- Expected performance
  - ~1 hour to call genotypes/SNP of one high coverage exome capture sequencing sample (~60X) using single CPU core