Alignment post-processing and variant calling

Simon Rasmussen
Center for Biological Sequence Analysis
Outline

① Introduction

② Alignment post-processing

③ Aligned reads (BAM) ➔ genomic variation (VCF)

④ Variant effect
Generalized NGS analysis

- Raw reads
- Pre-processing
- Assembly: Alignment / de novo
- Application specific: Variant calling, count matrix, ...
- Compare samples / methods
- Answer?

Data size
In principle it is very simple...
But reality is slightly more complex...

How much of what we observe is actually due to errors (noise) and how much represent real genomic variation?
Recommended workflow

1. Alignment sorting, filtering and indexing (.sort.flt.bam)
2. Local Realignment (.realign.bam)
3. Base Quality Score Recalibration (recalibrated.bam)
4. Duplicate marking/removal (.rmdup.bam)

Alignment post-processing

Alignment statistics

Mapping & alignment (.bam)

Variant calling & genotyping (.vcf)

Variant filtering

Variant effect

BAM ➔ VCF
ALIGNMENT POST-PROCESSING
Recommended workflow

1. Alignment sorting, filtering and indexing (sort.flt.bam)
2. Local Realignment (realign.bam)
3. Base Quality Score Recalibration (recalibrated.bam)
4. Duplicate marking/removal (rmdup.bam)

Alignment post-processing

Alignment statistics

Mapping & alignment (.bam)

BAM ➞ VCF

Variant calling & genotyping (.vcf)

Variant filtering

Variant effect
Recommended workflow

1. Alignment sorting, filtering and indexing (.sort.flt.bam)
2. Local Realignment (.realign.bam)
3. Base Quality Score Recalibration (recalibrated.bam)
4. Mapping & alignment (.bam)
5. Alignment post-processing
   - Duplicate marking/removal (.rmdup.bam)
   - Variant calling & genotyping (.vcf)
6. Alignment statistics
   - Variant filtering
   - Variant effect
Local realignment\textsuperscript{19,20}

- Each read is independently aligned to the reference genome to cut down on the computational cost.
- But this means that many reads spanning indels are likely to be misaligned.
- We see this as regions containing indels as well as clusters of mismatching bases.
Local realignment\textsuperscript{19,20} 

확장된 기능 

- Step 1: Identify regions likely in need of realignment.
- Step 2: Perform a multiple sequence realignment in this region, such that the number of mismatching bases is minimized across all reads.

1 – At least one read contains an indel.
2 – A cluster of mismatching bases exists.
3 – An already known indel segregates at the site.
Recommended workflow

1. Alignment sorting, filtering and indexing (.sort.flt.bam)
2. Local Realignment (.realign.bam)
3. Base Quality Score (recalibrated.bam)
4. Duplicate marking/removal (.rmdup.bam)
5. Alignment post-processing
6. Mapping & alignment (.bam)
7. Alignment statistics
8. Variant calling & genotyping (.vcf)
   - Variant filtering
   - Variant effect

BAM ➜ VCF
**Duplicate marking/removal**

Why do we have duplicates in our data?
- The PCR amplification step included in the majority of NGS library construction techniques can introduce duplicates in the data.

Why do we need to remove these?
- It will bias our variant calls.
- PCR errors are propagated and sampled many times giving rise to FP variant calls.

Basic concepts of duplicate marking algorithm:
1. Identify genomic position and strand for 5’-most bases.
2. Mark reads that are duplicates of each other.
3. Within a group of duplicate reads the read with the highest sum of base quality scores is retained.

But it is not perfect...
- Does not account for sequencing errors.
- Does not account for natural duplicates.
- Does not account for duplicate reads with different mapping locations.

Recommended workflow

Alignment statistics

- Mapping & alignment (.bam)
- Alignment sorting, filtering and indexing (.sort.flt.bam)
- Local Realignment (.realign.bam)
- Duplicate marking/removal (.rmdup.bam)

Alignment post-processing

- Base Quality Score (recalibrated.bam)

Variant calling & genotyping (.vcf)

- Variant filtering
- Variant effect
Base quality score recalibration\textsuperscript{2,3}

- Base-calling algorithms produce per-base quality scores by using noise estimates from image analysis.

- The raw Phred-scaled quality scores produced by base-calling algorithms may not accurately reflect the true base-calling error rates.

- Obtaining well-calibrated base quality scores is important as SNP and genotype calling depends on both the base calls and the per-base quality scores.
Base quality scores are systematically biased.

---

Underestimated

Overestimated

---

Accuracy (empirical - reported quality)

Machine cycle

Dinucleotide
Base quality score recalibration$^{2,3}$

• How it works!

1. Collect information regarding the following features of the bases:
   - Reported quality score
   - Position within the read (machine cycle)
   - Dinucleotide context (current base plus previous base)

2. Count the number of times a site was a mismatch to the reference (excluding known polymorphic sites).

3. Estimate new quality score as:

   \[
   \text{Phred-scaled quality score} = \frac{\text{# of reference mismatches} + 1}{\text{# of observed bases} + 2}
   \]

Example:

We observed A [AA, pos. 35, Q20] a 1,000,000 times.
A in this context mismatches the reference a 1,000 times.
This gives us: Q value $= -10 \log_{10}(\frac{1,000+1}{1,000,000+2}) \sim Q30$
FROM BAM TO VCF

Aligned reads → genomic variation
Recommended workflow

1. Alignment sorting, filtering and indexing (sort.flt.bam)
2. Local Realignment (realign.bam)
3. Base Quality Score (recalibrated.bam)
4. Duplicate marking/removal (rmdup.bam)
5. Alignment post-processing
   - Variant calling & genotyping (.vcf)
     - Variant filtering
     - Variant effect
Aims

- **Variant calling:** Identify polymorphic sites => sites that differs from the reference.
- **Genotyping:** Determine the genotype for a certain individual at such sites.

**Early methods**

- Works by simply counting the alleles at each site, and then identifying a variant by use of simple cutoff rules.
Identify variants

---TCGTCGTGGTTGAACGTACCGTTCCCTGAGGCTTTAT---

TCCTCGTGGTTGAACGGA
CGTCGTGGTTGAACGGAC
CGTGGTTGAACGGACGTAC
GTGGTTGAACGGACGTACCGTTCC
TGAACGGACGTACCGTTCC
GAACGGACGTACCGTTCCCTG
TGAACGGACGTACCGTTCC
GAACGGACGTACCGTTCCCTGAGGC
ACGGACGTACAGTTCCCT
GGACGTACCGTTCC
GTACCCTGAG--TTA
TCCCTGAG--TTA
CCTGAGGCTTTAT

What if it is human DNA?
What if it is bacterial DNA?
Variant calling & genotyping

- Several Bayesian genotyping methods available:
- Use the information on base counts, base qualities, mapping quality
- Calculate genotype likelihoods

Angsd

Samtools$^{21}$

UnifiedGenotyper (GATK)$^9$

FreeBayes$^{11}$

HaplotypeCaller (GATK)$^{10}$
Variant calling & genotyping

- Probabilistic methods – a simple Bayesian genotyper\(^5\)
- The posterior probability of each genotype given the pileup of sequence reads is given by Bayes' theorem:

\[
p(G|D) = \frac{p(G)p(D|G)}{p(D)}
\]

\[
p(D|G) = \prod_{b \in \text{pileup}} p(b|G)
\]

\[
p(b|G) = p(b\{|A_1, A_2\}) = \frac{1}{2} p(b|A_1) + \frac{1}{2} p(b|A_2)
\]

\[
p(b|A) = \begin{cases} 
\frac{e}{3} & : b \neq A \\
1 - e & : b = A 
\end{cases}
\]

- \(p(G|D)\): Prob. of genotype given data (posterior probability)
- \(p(G)\): Prior prob. based on allele freqs. from HW
- \(p(D|G)\): Prob. of data given genotype (genotype likelihood)
- \(p(D)\): Prob. of data
- \(p(b|G)\): Prob. of genotype given observed base at a site
- \(e\) is the Phred scaled quality score of the base
We see 3 reads: T (Q20), T (Q20), G (Q10)

1. Calculate Probability of each base given the observed allele $p(b|A)$:

For read 1 and 2: T (Q20)

$p(A|A) = 0.01/3$
$p(C|A) = 0.01/3$
$p(G|A) = 0.01/3$
$p(T|A) = 0.99$

For read 3: G (Q10):

$p(A|A) = 0.1/3$
$p(C|A) = 0.1/3$
$p(G|A) = 0.9$
$p(T|A) = 0.1/3$

Determine for each possible genotype
We see 3 reads: T (Q20), T (Q20), G (Q10)

2. Calculate prob. of each base given genotype $p(b \mid G)$:

For reads 1 and 2:
- TT: $0.99/2 + 0.99/2 = 0.99$
- TG: $0.99/2 + 0.01/(3\times2) = 0.49$
- GG: $0.01/(3\times2) + 0.01/(3\times2) = 0.003$

For read 3:
- TT: $0.1/(3\times2) + 0.1/(3\times2) = 0.03$
- TG: $0.1/(3\times2) + 0.9/2 = 0.47$
- GG: $0.9/2 + 0.9/2 = 0.9$
We see 3 reads: T (Q20), T (Q20), G (Q10)

3. Sum over all bases to calculate prob. of genotype $p(D \mid G)$:
   - TT: $0.99 \times 0.99 \times 0.03 = 0.029$
   - TG: $0.49 \times 0.49 \times 0.47 = 0.11$
   - GG: $0.003 \times 0.003 \times 0.9 = 8e-06$

4. Calculate posterior prob. $p(G \mid D)$:
   
   $p(D) = 0.029 + 0.11 + 8e-06 = 0.139$
   
   $p(G)$: for simplicity we assume uniform prior
   
   Posterior probability $p(G \mid D)$:
   - TT: $0.029/p(D) = 0.21$
   - TG: $0.11/p(D) = 0.79$
   - GG: $8e-06/p(D) = 5.8e-05$

This is “GQ” in VCF
Recommended workflow

1. Mapping & alignment (.bam)
2. Alignment sorting, filtering and indexing (.sort.flt.bam)
3. Local Realignment (.realign.bam)
4. Duplicate marking/removal (.rmdup.bam)
5. Base Quality Score (recalibrated.bam)

BAM → VCF

- Variant calling & genotyping (.vcf)
- Variant filtering
- Variant effect
Variant filtration (soft)$^{18}$

- **Variant quality score recalibration (GATK)**
  - How do we remove false positive calls?

- Use known polymorphic sites to estimate what a real variant and a false variant “looks like”
  - Learn how does the known sites (=truth set) look like in our data
  - Examples can be: quality/depth, MQ, BaseReadPos, strand bias
  - Evaluate on all our data, filter sites that look different!
Variant filtration (soft)\textsuperscript{18}

The raw SNP calls are partitioned into quality trances based on truth sensitivity.

Truth: Accept all sites until X\% of the truth sites have been found.
Variant filtration (hard)

• Hard filtering\(^{16}\) (when VQSR is not possible*)

  – Variant quality score /depth
  – Mapping quality
  – Strand bias (the variant being seen only on the forward strand or only on the reverse strand)
  – Depth

• Some recommendations\(^{17}\)

For SNPs:
- \(\text{QD} < 2.0\)
- \(\text{MQ} < 40.0\)
- \(\text{FS} > 60.0\)
- \(\text{HaplotypeScore} > 13.0\)
- \(\text{MQRankSum} < -12.5\)
- \(\text{ReadPosRankSum} < -8.0\)

For indels:
- \(\text{QD} < 2.0\)
- \(\text{ReadPosRankSum} < -20.0\)
- \(\text{InbreedingCoeff} < -0.8\)
- \(\text{FS} > 200.0\)

\(^{16}\) number of samples < 30 or if you're doing targeted resequencing of a small region, non-model organism.
Recommended workflow

1. **Alignment sorting, filtering and indexing** (.sort.flt.bam)
2. **Local Realignment** (.realign.bam)
3. **Base Quality Score** (recalibrated.bam)
4. **Duplicate marking/removal** (.rmdup.bam)
5. **Alignment post-processing**
   - **Mapping & alignment** (.bam)
6. **Variant calling & genotyping** (.vcf)
7. **Variant filtering**
8. **Variant effect**

Alignment statistics:
- **Alignment statistics**
  - **Mapping & alignment** (.bam)
  - **Alignment sorting, filtering and indexing** (.sort.flt.bam)
  - **Local Realignment** (.realign.bam)
  - **Duplicate marking/removal** (.rmdup.bam)
  - **Base Quality Score** (recalibrated.bam)

BAM → VCF
Variant annotation

Variant Effect Predictor script

Download

The Variant Effect Predictor script can be downloaded as a tarball from the Ensembl CVS server:

Download latest version (71)

NOTE: VEP version numbers have from release 71 changed to match Ensembl release numbers.

It is also included as part of the ensembl-tools module of the Ensembl API - you can find it in the ense

Previous versions: Show

Requirements

Version 71 of the script requires at least version 71 of the Ensembl Core and Variation APIs and their:

the supplied installer script.

Link: http://www.ensembl.org/info/docs/variation/vep/vep_script.html
References

2) http://gatkforums.broadinstitute.org/discussion/44/base-quality-score-recalibration-bqsr
3) http://www.youtube.com/watch?v=L4D1dwES9s8
7) http://www.1000genomes.org/wiki/Analysis/Variant%20Call%20Format/vcf-variant-call-format-version-41
8) http://www.ensembl.org/info/docs/variation/vep/index.html
References

9) http://www.broadinstitute.org/gatk/gatkdocs/org_broadinstitute_sting_gatk_walkers_genotyper_UnifiedGenotyper.html
10) http://www.broadinstitute.org/gatk/gatkdocs/org_broadinstitute_sting_gatk_walkers_haplotypecaller_HaplotypeCaller.html
12) http://www.broadinstitute.org/gatk/gatkdocs/org_broadinstitute_sting_gatk_walkers_variantrecalibration_VariantRecalibrator.html
13) http://www.broadinstitute.org/gatk/gatkdocs/org_broadinstitute_sting_gatk_walkers_variantrecalibration_ApplyRecalibration.html
14) http://www.broadinstitute.org/gatk/gatkdocs/org_broadinstitute_sting_gatk_walkers_varianteval_VariantEval.html
15) http://www.broadinstitute.org/gatk/guide/article?id=1247
16) http://www.broadinstitute.org/gatk/gatkdocs/org_broadinstitute_sting_gatk_walkers_filters_VariantFiltration.html
17) http://www.broadinstitute.org/gatk/guide/article?id=1186
18) http://www.broadinstitute.org/gatk/guide/article?id=39
19) http://www.broadinstitute.org/gatk/gatkdocs/org_broadinstitute_sting_gatk_walkers_indels_RealignerTargetCreator.html
20) http://www.broadinstitute.org/gatk/gatkdocs/org_broadinstitute_sting_gatk_walkers_indels_IndelRealigner.html
21) Heng Li, Jue Ruan, Richard Durbin. Mapping short DNA sequencing reads and calling variants using mapping quality scores, Genome Research, 2008