Chapter 6

Short Read Mapping for Exome Sequencing

Xueya Zhou, Suying Bao, Binbin Wang, Xuegong Zhang, and You-Qiang Song

Abstract

Mapping short reads to the reference genome is very often the prerequisite for applications utilizing the next-generation sequencing technologies. A dozen of software tools developed for this purpose have been widely used. But many practical issues remained when utilizing them to build a computational pipeline for downstream analyses. In this chapter, we describe the read mapping procedures adopted in our lab for the exome sequencing studies as an example to illustrate those practical details.

Key words Next-generation sequencing, Short read mapping, Exome sequencing, Burrows–Wheeler aligner

1 Introduction

The past few years saw the application of next-generation sequencing technologies in many areas of biomedical research, including medical resequencing and variant discovery [1], mRNA expression profiling and novel RNA discovery [2], protein-binding site identification [3], etc. All new sequencing platforms (Illumina, ABI SOLID, and Roche/454) were able to produce giga base pairs of raw sequences per machine day, but individual output of the machines (or short reads) can never make sense until their genomic sources are known. In the presence of reference genome assembly, quickly identifying the most likely position where short reads originated is the first and foremost step in data analysis. Given that traditional alignment tools (e.g., [4]) were not capable to carry out this task efficiently, a plethora of software dedicated for short read mapping has been developed to meet the challenge. Despite their implementation details, most of the tools can be broadly classified into two categories: one used hash tables to index either reference or reads; the other was based on Burrows–Wheeler transformation of the reference for rapid searching. Since the exact solution of searching all possible places

Noam Shomron (ed.), *Deep Sequencing Data Analysis*, Methods in Molecular Biology, vol. 1038,
DOI 10.1007/978-1-62703-514-9_6,

of a read is not computationally tractable, all the tools first used heuristic techniques to quickly identify a small set of possible locations, and then run more accurate algorithms on those candidate locations to refine the best mapping positions. The algorithmic principles have been reviewed by several authors (e.g., [5, 6]). Generally speaking, different tools represent different trade-offs between speed and accuracy. They also differ in ways that different features are accommodated (summarized in Table 1).

When applying theories into practices, however, we are more frequently faced with issues like choosing a software tool, adjusting parameters, manipulating input and output files, and performing quality controls. While previous work by us and others that focused on comparison and benchmark can address some of the questions [7, 8], most have never been summarized in the literature. To fill in this gap, we share readers with our read mapping procedures adopted in exome sequencing studies, and use them as an example to illustrate solutions to many practical issues. Although many steps are specific to exome sequencing data, our discussions are generalized to other areas as well. We do not cover the read mapping problem for RNAseq and methylation profiling, because specialized software pipelines (e.g., [9, 10]) have been developed for them.

2 Materials

To carry out the steps in our protocol, a computer workstation running Unix-based operation system is required. The procedures listed in Subheading 3 have been tested on a workstation with 8 Cores (Intel Xeon 2.4GHz) and 16 GB RAM running Redhat Enterprise Linux v5.0. At least 200 GB free disk space was needed for storing data files. Software tools and data sets used in the procedures are listed below.

2.1 Software

1. FastQC (v0.10.1) was used to assess the read quality, http://www.bioinformatics.babraham.ac.uk/projects/fastqc/.
2. BWA (v0.5.9) was used for read mapping (*see* **Note 1**), http://bio-bwa.sourceforge.net/.
3. picard (v1.6.7) was used to manipulate alignment files and generate quality control statistics, http://picard.sourceforge.net/.
4. verifyBamID (2012.06.20 version) was used to test sample contamination, http://genome.sph.umich.edu/wiki/VerifyBamID.
5. GATK (v1.4-9-g1f1233b) was used to post-process the alignment files, http://www.broadinstitute.org/gsa/wiki/index.php/Home_Page.
6. SAMtools (v0.1.8), http://samtools.sourceforge.net/.

Table 1
An incomplete survey of tools for mapping short reads

Tools	Web sites	Mapping strategy	Gapped alignment?	Color space?	Roche 454?	Use base quality?[a]	References
BWA	http://bio-bwa.sourceforge.net/	BWT based	Y	Y	SE only[b]	N	[17]
Bowtie	http://bowtie-bio.sourceforge.net	BWT based	N[c]	Y	N	Y	[19]
BFAST	http://sourceforge.net/apps/mediawiki/bfast	Hash the genome	Y	Y	Y	N	[20, 21]
Mosaik	http://code.google.com/p/mosaik-aligner/	Hash the genome	Y	Y	Y	N	–
mrFAST/mrsFAST	http://mrsfast.sourceforge.net/	Hash the genome	N	Y[d]	N	N	[13]
Novoalign	http://www.novocraft.com/	Hash the genome	Y	Y	Y	Y	–
SHRiMP	http://compbio.cs.toronto.edu/shrimp/	Hash the reads	Y	Y	Y	N	[23, 24]
SOAP2	http://soap.genomics.org.cn/	BWT based	PE only[e]	Y	N	N	[25]
Stampy	http://www.well.ox.ac.uk/project-stampy	Hash the genome	Y	Y	N	Y	[12]
SSAHA2	http://www.sanger.ac.uk/resources/software/ssaha2/	Hash the genome	Y	Y	Y	N	[4]

[a]Use Base Quality refers to that the software explicitly uses base quality in calculating mapping quality or weighting mismatches
[b]*SE* single-end sequencing
[c]Bowtie v2 [18] now supports gapped alignment
[d]A sister program drFAST [22] of the same algorithm family was designed to work on color space
[e]*PE* paired-end sequencing

7. BEDtools (v2.16.2), http://code.google.com/p/bedtools/.
8. VCFtools (v0.1.9), http://vcftools.sourceforge.net/.

2.2 Data Files

1. The latest human genome reference sequences (*see* **Note 2**): ftp://ftp.ncbi.nih.gov/1000genomes/ftp/technical/reference/human_g1k_v37.fasta.gz.
2. The example exome sequencing data for sample NA20322 sequenced by the 1000 Genomes Project: ftp://ftp-trace.ncbi.nih.gov/1000genomes/ftp/data/NA20322/sequence_read/. The following files are the cleaned sequencing reads of NA20322's exome generated by two lanes of Illumina GAII pair-end sequencing (*see* **Note 3**): SRR070556_1.filt.fastq.gz, SRR070556_2.filt.fastq.gz, SRR070839_1.filt.fastq.gz, SRR070839_2.filt.fastq.gz.
3. Definition files for the targets and baits of Agilent SureSelect all exons V2 kit (*see* **Note 4**): https://earray.chem.agilent.com/earray/. An account should be registered before log in the eArray system. You can find detailed information about this kit version either by browsing the library or by searching ELID: S0293689. Download both the ANNOTATION (targets) and BED (baits) files.
4. Download the resource bundles of Genome Analysis Toolkit (GATK): http://www.broadinstitute.org/gsa/wiki/index.php/GATK_resource_bundle. We only used resource bundle version 1.5 for b37. Each data file should be decompressed after downloading.

3 Methods

We provide all steps as UNIX bash shell instructions. Long commands were separated into multiple lines. The whole procedures take ~1 day to finish.

3.1 Preparing Input Files

1. Create the following data directories under the current working directory, and copy corresponding data files into them: genome, containing the downloaded genome reference sequences; gatk, containing the GATK resource bundle for b37; exome, containing the kit definitions with two subdirectories ANNOTATION and BED; fastq, containing the downloaded fastq files for sample NA20322; jar, containing the Java archive files of picard and GATK packages.
2. Use maskFastaFromBed from BEDtools to mask out pseudoautosomal regions (PARs) on chromosome Y in the reference sequences (*see* **Note 2**).

```
guzip genome/human_g1k_v37.fasta.gz
cat <<EOF | awk 'BEGIN{OFS="\t"}{print $1,$2,$3,
$4}' >
/tmp/mask.bed
Y 10001 2649520 PAR1
Y 59034050 59363566 PAR2
EOF
maskFastaFromBed -fi genome/human_g1k_v37.fasta
-bed /tmp/mask.bed \
  -fo genome/b37.fasta
```

3. Run BWA to index the masked genome reference (*see* **Note 5**).

   ```
   bwa index -p genome/b37 -a bwtsw genome/b37.fasta
   ```

   You will find the following eight files created in the genome folder: b37.amb, b37.ann, b37.bwt, b37.pac, b37.rbwt, b37.rpac, b37.rsa, b37.sa.

4. Create FASTA file index and dictionary for the reference.

   ```
   samtools faidx genome/b37.fasta
   java -Xmx2g -jar jar/CreateSequenceDictionary.jar \
     REFERENCE=genome/b37.fasta OUTPUT=genome/b37.
   dict GENOME _ASSEMBLY=b37
   ```

   It will result in two files, b37.fasta.fai and b37.fasta.dict, that will be used by some picard programs.

5. Run FastQC to evaluate the read quality and determine the file type and base quality score encoding (*see* **Note 6**).

   ```
   mkdir fastqc
   fastqc --outdir fastqc/ fastq/*.gz
   ```

   The report for each FASTQ file is reformatted in HTML file under the fastqc directory. The "Basic Statistics" section shows that our data files contain conventional base calls with qualities encoded in Sanger scheme. The report also shows the per-base quality, sequence duplication levels among others (*see* **Note 7**).

6. Convert target and bait definitions to picard interval list format (*see* **Note 8**).

   ```
   awk    'BEGIN{OFS="\t"}NR>1{gsub("chr","",$1);
   print $0}' \
   exome/ANNOTATION/SureSelect_All_Exon_V2_-
   with_annotation.hg19.bed \
     > exome/targets.bed
   cat genome/b37.dict > exome/targets.interval_list
   awk 'BEGIN{OFS="\t"}NR>1{print $1,$2+1,$3,"+",
   $4}' \
     exome/targets.bed >> exome/targets.interval_
   list
   ```

```
cat genome/b37.dict > exome/baits.interval_list
awk  'BEGIN{OFS="\t"}{gsub("chr","",$1);  print
$1,$2+1,$3,$6,$4}' \
  exome/BED/029368_D_BED_20111101.bed \
  >> exome/baits.interval_list
```

3.2 Mapping Reads to Reference

1. Generate alignments in suffix array coordinates for each end separately, using six parallel threads to speed up (*see* **Note 9**).

```
for RUN in SRR070556 SRR070839; do
  mkdir -p mapping/$RUN
  bwa aln -t 6 -q 10 -f mapping/$RUN/${RUN}_1.sai
genome/b37 \
  fastq/$_1.filt.fastq.gz
  bwa aln -t 6 -q 10 -f mapping/$RUN/${RUN}_2.sai
genome/b37 \
  fastq/$_2.filt.fastq.gz
done
```

2. Create alignment in SAM format for each run of paired-end reads (*see* **Note 10**).

```
for RUN in SRR070556 SRR070839; do
  bwa sampe -f mapping/${RUN}/$RUN.sam \-r
"@RG\tID:$RUN\tSM:NA20322\tLB:NA20322\tPL:
ILLUMINA" genome/b37 \
  mapping/${RUN}/${RUN}_1.sai
mapping/${RUN}/${RUN}_2.sai \
  fastq/${RUN}_1.filt.fastq.gz
fastq/${RUN}_2.filt.fastq.gz
done
```

3. Fix mate information in SAM output and convert to binary format (*see* **Note 11**).

```
for RUN in SRR070556 SRR070839; do
  java -XX:ParallelGCThreads=2 -Xmx2g -jar
jar/FixMateInformation.jar \
  INPUT=mapping/$RUN/$RUN.sam
OUTPUT=mapping/$RUN/$RUN.fixed.bam \
TMP_DIR=mapping/$RUN/tmp
VALIDATION_STRINGENCY=SILENT
done
```

4. Reorder the contigs and sort aligned reads by their mapped coordinates within contigs (*see* **Note 12**).

```
for RUN in SRR070556 SRR070839; do
  java -XX:ParallelGCThreads=2 -Xmx2g -jar jar/
ReorderSam.jar \
  INPUT=mapping/$RUN/$RUN.fixed.bam \
  OUTPUT=mapping/$RUN/$RUN.reordered.bam \
  REFERENCE=genome/b37.fasta
```

```
TMP_DIR=mapping/$RUN/tmp \
  VALIDATION_STRINGENCY=SILENT
java -XX:ParallelGCThreads=2 -Xmx2g -jar
jar/SortSam.jar \
  INPUT=mapping/$RUN/$RUN.reordered.bam \
  OUTPUT=mapping/$RUN.sorted.bam
TMP_DIR=mapping/$RUN/tmp \
  SORT_ORDER=coordinate CREATE_INDEX=true \
  VALIDATION_STRINGENCY=SILENT
Done
```

This step also creates index file (.bai) for each sorted BAM file.

5. Merge two lane-level alignments into the sample level.

```
java -XX:ParallelGCThreads=2 -Xmx2g -jar
jar/MergeSamFiles.jar \
  INPUT=mapping/SRR070556.sorted.bam
INPUT=mapping/SRR070 839.sorted.bam \
OUTPUT=mapping/NA20322.merged.bam    CREATE_IN-
DEX=true \
SORT_ORDER=coordinate USE_THREADING=true \
TMP_DIR=mapping/tmp   VALIDATION_STRINGENCY=SI-
LENT
```

The NA20322.merge.bam is sorted and index BAM file for the sample.

3.3 Post-processing on Aligned Reads

1. Mark up duplicated fragments detected from aligned reads (*see* **Note 13**).

```
mkdir sam_qc
java -XX:ParallelGCThreads=2 -Xmx2g -jar jar/
MarkDuplicates.jar \
  INPUT=mapping/NA20322.merged.bam
OUTPUT=mapping/NA20322.dupmarked.bam \
CREATE_INDEX=true
METRICS_FILE=sam_qc/NA20322.duplication_
metrics \
  TMP_DIR=mapping/tmp VALIDATION_STRINGENCY=SI-
LENT
```

This step also generates duplication metrics in sam_qc directory.

2. Realignment around indels using GATK (*see* **Note 14**). This is done by first generating targets for realignment, and then performing realignment within target intervals.

```
java -XX:ParallelGCThreads=2 -Xmx4g -jar jar/
GenomeAnalysisTK.jar \
  -l INFO -et STDOUT -T RealignerTargetCreator \
  -R gatk/b37.fasta -L exome/targets.interval_
  list \
```

```
  -I mapping/NA20322.dupmarked.bam \
  --known gatk/1000G_phase1.indels.b37.vcf \
  --known
gatk/Mills_and_1000G_gold_standard.indels.b37.
vcf \
  -o mapping/NA20322.realign.interval_list
java -XX:ParallelGCThreads=2 -Xmx4g -jar jar/
GenomeAnalysisTK.jar \
  -l INFO -et STDOUT -T IndelRealigner \
  -R gatk/b37.fasta -I mapping/NA20322.dupmarked.
  bam \
  -o mapping/NA20322.realigned.bam \
  -targetIntervals
mapping/NA20322.realign.interval_list \
  --knownAlleles gatk/1000G_phase1.indels.b37.
  vcf \
  --knownAlleles    gatk/Mills_and_1000G_gold_
  standard.indels.b37.vcf \
  --consensusDeterminationModel USE_READS
```

3. Recalibrate base quality scores (*see* **Note 15**). This is done by first tabulating the effect estimates of covariates at mismatches outside known SNPs, and then applying corrections over all base calls.

```
java -Xmx4g -XX:ParallelGCThreads=2 -jar jar/
GenomeAnalysisTK.jar \
  -l INFO -et STDOUT -T CountCovariates -nt 4 \
  -R gatk/b37/b37.fasta -knownSites
gatk/b37/dbsnp_135.b37.vcf \
  -I mapping/NA20322.realigned.bam --default_
platform Illumina \
  -cov ReadGroupCovariate -cov QualityScoreCov-
ariate \
  -cov CycleCovariate -cov DinucCovariate \
  -recalFile mapping/NA20322.pre_recal.csv
java -Xmx4g -XX:ParallelGCThreads=2 -jar jar/
GenomeAnalysisTK.jar \
  -l INFO -et STDOUT -T TableRecalibration \
  -R gatk/b37/b37.fasta --default_platform Illu-
  mina \
  -I mapping/NA20322.realigned.bam -o mapping/
  NA20322.recal.bam \
  -recalFile mapping/NA20322.pre_recal.csv
```

The resulting NA20322.recal.bam file is now suitable for variant calling. Before proceeding to downstream analysis, we need to ensure the data quality at first.

3.4 Quality Control on Alignment

1. Collect various quality control statistics from processed alignment file.

```
java -XX:ParallelGCThreads=2 -Xmx2g -jar jar/
CollectMultipleMetrics.jar \
  INPUT=mapping/NA20322.recal.bam
OUTPUT=sam_qc/NA20322 \
  REFERENCE_SEQUENCE=exome/b37.fasta \
  PROGRAM=CollectAlignmentSummaryMetrics \
  PROGRAM=QualityScoreDistribution \
  PROGRAM=MeanQualityByCycle \
  PROGRAM=PROGRAM=CollectInsertSizeMetrics \
  VALIDATION_STRINGENCY=SILENT
```

   It calculates the alignment statistics, insert size distribution, quality score distributions, and mean quality score for each machine cycle. The main results for our sample NA20322 are shown in Fig. 2a–c (*see* **Note 16**).

2. Evaluate the depth of coverage over target regions.

```
java -XX:ParallelGCThreads=2 -Xmx2g -jar jar/
CalculateHsMetrics.jar \
  INPUT=mapping/NA20322.recal.bam \
  OUTPUT=sam_qc/NA20322.hybrid_selection_-
  metrics \
  BAIT_INTERVALS=exome/baits.interval_list \
  TARGET_INTERVALS=exome/targets.interval_list \
PER_TARGET_COVERAGE=sam_qc/NA20322.per_tar-
get_coverage \
  REFERENCE_SEQUENCE=exome/b37/b37.fasta VALI-
DATION_STRINGENCY=SILENT
```

   It calculates various statistics evaluating the exome capture performance and sequencing completeness (shown in Fig. 2a), and also the depth of coverage for each target in a separate file.

3. We also need to test if the sequenced sample matches his/her own identity. The 1000 Genomes Project has genotyped all of the phase 1 samples on Illumina Omin2.5 SNP array. The genotypes in VCF file are a part of GATK bundle. The sample information can be downloaded from the Project FTP,[1] and copied to the k1g directory. The sample NA20322 belongs to ASW population, so we first extract SNP genotypes for ASW population within the autosomal targets of the exome kit.

[1]The sample information for the phase 1 of 1,000 Genomes Projects can be downloaded from ftp://ftp-trace.ncbi.nih.gov/1000genomes/ftp/release/20110521/phase1_integrated_calls.20101123.ALL.panel.

```
awk '$2=="ASW"{print $1}' k1g/phase1_integrated_calls.20101123.ALL.panel \
> k1g/ASW.keep
awk '$1 ~ /^[0-9]+$/' exome/targets.bed > exome/targets.auto.bed
vcftools --vcf gatk/1000G_omni2.5.b37.vcf --keep k1g/ASW.keep \
  --remove-filtered-all --bed exome/targets.auto.bed \
  --recode --out k1g/ASW.exome
```

4. Then run verifyBamID to detect sample swapping or contamination.

```
verifyBamID --vcf k1g/ASW.exome.recode.vcf \
  --bam mapping/NA20322.recal.bam --out sam_qc/idcheck --verbose \
  --maxDepth 2000 --precise --minQ 20 --maxQ 100 --minMapQ 17
```

The results at sample level can be found in file idechek.selfSM. Both CHIPMIX and FREEMIX statistics are less than 0.001 (*see* **Note 17**). It suggests that our sample matched to the identity, and there is no evidence of contamination from other samples.

4 Notes

1. We choose BWA for mapping reads in exome sequencing based on the following considerations. First, the medical resequencing projects require reads to be mapped in a way that maximizes the sensitivity for variant discovery. They were typically implemented with paired-end sequencing with ~100 bp length. For this purpose, gapped alignment supported by BWA is essential [11]. It was also shown to have higher accuracy and sensitivity than other tools as compared by Bao et al. [7]. Second, BWA supports the major features of the two widely used platforms for exome sequencing: Illumina and SOLiD. Both platforms typically have higher error rate at 5′ end; and SOLiD system also provides error-correctable strategy. Third, we also need to strike a balance between accuracy and efficiency. For example, stampy and novoalign were both shown to have lower mapping errors than BWA, but their throughputs were also an order of magnitude lower [12]. A typical human exome contains over 10 giga base pairs, which will take ~10 CPU days to align using stampy but less than one day using BWA running four parallel threads. Although the speed may be less an issue if you have abundant computational resources, previous studies also showed that increased accuracy like novoalign did not show

much improved performance in SNP calling as compared with mapping with BWA [5]. So we considered BWA as a reasonable choice. But the effect of mapping accuracy on indel calling is less clear and remains to be investigated in the future.

The choice of read mapping software is generally influenced by the factors like the biological question, the sequencing platform, and trade-offs between speed and performance. In comparative genomics, when mapping reads to a diverged species, tools that retain sensitivity given higher rate of mismatches should be considered. In applications like ChIP-seq, reads with typical length of ~50 bp were generated; the subsequent analysis focuses on the peak calling from aligned read depth; then gapped alignment may not be essential, and priorities can be given to run time efficiency. Tools differ in their behavior in handing repetitive sequences. mrsFAST is the only tool to report all possible mapping positions given the number of mismatches, which is critical for detecting structural variation [13]. Different tools also have different support for reads generated by Roche 454 platform, which are of intermediate length (200–400 bp) and have an increased indel error in the presence of homopolymers. All the above considerations need to be synthesized in choosing a read mapping tool.

2. In our case of human genetics study, it is recommended to use the reference sequence adopted by the 1000 Genomes Project. This version was mainly based on the latest human genome reference sequences maintained by the Genome Reference Consortium (GRC). The 1000 Genomes version of the reference contained all 24 assembled chromosomes (1–22, X, Y) and 59 unplaced contigs (collectively referred to as the primary assembly) of the GRCh37 release, and replaced the mitochondrion sequence with the revised Cambridge Reference Sequences (Genbank: NC_012920). We often casually refer to the same version as the hg19 in UCSC's genome browser, which was also derived from GRCh37. The sequences on the primary assemblies are the same, except that the UCSC has its own naming conventions on chromosome and contigs. But the UCSC used the old version of mitochondrion (Genbank: NC_001807) and also included alternative representations of nine genomic loci like MHC. Because those alternative loci contain long flanking sequences highly homologous to the primary assembly, we do not include them in our reference; otherwise, it will artificially reduce mapping qualities for reads mapped to those positions. Although we used human as example, the general principle for choosing a reference is the same for other species: the reference sequence should be the *best known haploid representation* of a genome from which the short reads were presumed to be generated. The unlocalized and unplaced contigs should also be included, but not

alternative haplotypes of known loci. We also noted that two PARs on the chromosome X and Y were redundant to each other, so one copy of them will be masked out before indexing the sequences (*see* **step 2** in Subheading 3.1).

3. The 1000 Genomes Project kept updating the raw data regularly on two of FTP mirror sites at EBI (ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/) and NCBI (ftp://ftp-trace.ncbi.nih.gov/1000genomes/ftp/). Information about each data file in the project was recorded and updated in the sequence index files (can be found under sequence_indices directory). From that file we can learn that NA20322 was sampled from a population of African Ancestry in Southwest USA. The exome of NA20322 was sequenced in Washington University Genome Sequencing Center (WUGSC) on two lanes of Illumina Genome Analyzer II. When samples were sequenced on multiple lanes, we suggest to treat the raw sequences separately (*see* also **Note 10**). The exomes sequenced in WUGSC were captured using Agilent SureSelect V2 kit.[2]
4. Baits are the nomenclature of Agilent Technology; they are biotinylated 170mer RNA capture probes. Exome enrichment kits from other manufacturers used DNA molecules with different lengths as the capture probes. Their information can be found on the manufacturers' Web sites. The downloaded target and bait definition files are in BED format.[3] Each row represents a target or a bait interval on the UCSC's hg19 genome coordinates. It should be noted that the Agilent V2 exome kit only included target regions on chromosomes 1–22, X, and Y; so it is equivalent to the 1000 Genomes version of reference after stripping out the chromosome name prefix "chr." The first three columns give the chromosome name, and start and end positions. The intervals are zero-based half open.
5. The -a switch tells the program to index the genome using bwtsw algorithm, which is the only option for the size of human genome. For mapping reads generated by SOLID platform, the index should be generated by adding -c option. Under color space mode, three additional index files will be generated: b37.nt.amb, b37.nt.ann, and b37.nt.pca. Please also be noted that the support for color space has been discontinued after BWA v0.6.0.
6. The FastQ format is most widely used to store short reads. Most sequencing platforms can convert their output files to FastQ format. It contains information for one read every four lines. The first line always starts with "@" and specifies the name

[2]Information about exome targets of 1,000 Genomes Projects can be found in the following documents on the Project FTP: technical/reference/exome_pull_down_targets/README.20120518.exome.consensus.

[3]UCSC BED format is documented in http://genome.ucsc.edu/FAQ/FAQformat#format1.

of the read; the third line contains either only "+" or followed by the read name starting after "@" in the first line. Paired-end (PE) reads are stored in a pair of files, with reads kept in the same sorted order in both files. The corresponding read names must be identical up to the tailing "/1" or "/2." In PE sequencing, reads are generated from both ends of a captured DNA fragment with known lengths; such information can be used to constrain the mapping of both ends. Reads are paired based on their names during the mapping process. When read names contain white spaces, only the part before first white space will be used by the mapping tools.

The second line contains the actual sequence of the read. They can be either in bases or in color space. The base qualities are encoded as ASCII characters in the last line. The base qualities are Phred scaled (ten times the minus log10) error probabilities with some added offsets. There are three different quality encodings for FastQ files [14]. The traditional Sanger scheme used ASCII characters 33–126 to encode qualities from 0 to 93. The Solexa score has now been considered as legacy and rarely used these days. FastQ files produced by the earlier version of Illumina (Illumina 1.3+) used Phred score 0–62 encoded in ASCII 64–126. There is also a subversion of FastQ (Illumina 1.5+), which is the same as Illumina 1.3+, but do not use ASCII 64,65, and ASCII 66 was reserved to mean "the base should not be considered in downstream analysis". The latest format (Illumina1.8 or later) has switched back to the Sanger standard. Most mapping tools assume by default that the quality scores are in Sanger encoding scheme. Mapping software like BWA also support Illumina 1.3+ scheme as a command line option. In case a quality score is not supported, FastQ format conversion can be achieved by several tools (e.g., fq_all2std.pl script in MAQ,[4] NGSQC Toolkit[5]). Incorrect specification of base quality encoding will result in erroneous mapping if the tool used base quality score to calculate the mapping quality. It will also affect the downstream analysis (e.g., variant calling) when quality score is incompatible with the presumed encoding.

7. The FastQC's report shows a summary of each analysis module and indications of whether the results are normal or unusual. Some optional preprocessing steps can be taken to filter out reads with low average base qualities, or to trim out low-quality bases at 5′ ends. We skipped these steps in our projects, because filtering step has been done by the Illumina base calling pipelines, and BWA contains option to automatically trim low-quality bases at the ends. And the example FastQ used here

[4]MAQ can be downloaded from http://maq.sourceforge.net/.
[5]NGS QC Toolkit: http://www.nipgr.res.in/ngsqctoolkit.html.

has been filtered by the 1000 Genomes Consortium. If the preprocessing steps were to be taken, care must be taken to preserve paired-end information in FastQ files (*see* also **Note 6**).

8. Picard interval list format contains sequence dictionary in the header followed by one line per interval.[6] The position intervals are one-based, different from UCSC's BED format.
9. BWA's default parameter setting works well for aligning exome sequencing reads. We discuss some options here, because many of them are shared by similar ones in other tools, and you may tweak them in other applications.
   (a) Mismatches: BWA accepts or rejects an alignment based on the counted number or fraction of mismatches between read and the genomic position specified by -n option. Other tools may also use the sum of quality scores at mismatch positions. The default parameters are tuned for genomes with low polymorphisms.
   (b) Seeding: Seed refers to the first few tens of base pairs of a read. This part of a read is expected to contain fewer errors. Many tools used seed in their heuristic search that maximizes the performance. BWA's -l and -k options were designed for seeding strategy, but they are disabled by default.
   (c) Color space: For reads generated by SOLiD systems, all except first base are given a number out of four depending on the value of previous base. Mapping in color space directly has the advantage of distinguishing sequencing errors from true polymorphism. BWA's -c option enables this function, given reads in color spaces.
   (d) Base trimming: BWA contains a -q option to trim low-quality bases at 5′ end. Its meaning is explained in Fig. 1.
   (e) Quality score: BWA assumes that the quality score is Sanger encoding; use -I if the base quality is in Illumina 1.3+ encoding.
10. BWA separates the index searching and reads alignment in two steps. When paired-end reads are used, the distribution of fragment length is estimated from the data, unless there are too few reads. In the latter case, users can specify the minimal and maximum insert sizes. Most other tools have a similar behavior.

    It is important in this step to specify the read group information by -r option. This line will be put in the generated SAM file specifying the identity for the read group, library, and sample. The read group refers to a set of reads that are generated by a single lane from a single sequencing run. The reads

[6]Picard interval list format is explained in http://www.broadinstitute.org/gsa/wiki/index.php/Input_files_for_the_GATK#Intervals.

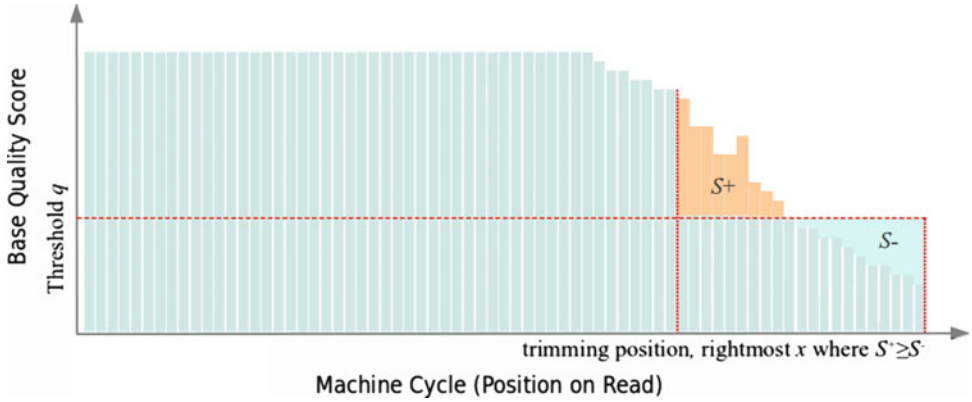

Fig. 1 Explaining BWA's read trimming option. Shown in the figure is a schematic view of quality scores along the positions of a read. Given a threshold *q*, if there are positions at the 5′ end having quality score less than *q*, BWA will determine the trimming bases from the leftmost positions where area shown in *orange* is no less than areas shown in *cyan*

within the same read group will have similar error profiles but may differ systematically from reads in other read group. This information will be used in quality score recalibration (*see* also **Note 15**). The library tag (LB) is also critical for marking up duplicated reads (*see* also **Note 13**). If a mapping tool did not accept read group information, we can also add read group information to the header in output file using ReplaceSamHeader program in picard package.

11. The BWA's alignment output is in SAM format,[7] which has become the standard for data exchange [15]. Most other mapping tools can either directly support SAM output, or provide auxiliary scripts to convert their native outputs to SAM format. For reduced storage, the SAM files are usually compressed in a binary format (BAM), and can be sorted and indexed in a way that facilitates range query.

    It should be emphasized that SAM files are software dependent. In addition to different mapping accuracies, software also have their own idiosyncrasies in outputting SAM file formats. For example, the SAM specification required that the record for one read contains some information about its mate pair which is also stored in the record for its mate. Although it seemed redundant, it can be helpful in many cases. This redundancy also created some inconsistency usually found in BWA's output. Broken mate information has nothing to imply about the mapping qualities; it just means that the software that aligned these reads did not set the information correctly. The FixMateInformation program will attempt to fill in these attributes.

[7]Details can be found in http://samtools.sourceforge.net/SAM1.pdf.

There are also some discrepancies to the SAM standard that cannot be corrected by the picard tools. For example, BWA concatenates reference contigs together before indexing; if a read happened to (erroneously) map to the position spanning the edge of one contig onto another, it will have a mapping position and quality but marked as unmapped. When the file is parsed by picard, it triggers an "MAPQ must should be 0 for unmapped read" error, since picard is very picky about the file format. As we believe that BWA's SAM output should be correct, it is no harm to left as is. So we stop the error checking by setting the picard command line option VALIDATION_STRINGENCY=SILENT. Picard package also comes with a utility program ValidateSamFile that can be used to check if a SAM file strictly conforms to the standard.

Picard determines whether to write in SAM or BAM format by examining the file extension of the output file. The format conversion is done by giving appropriate file extension.

12. Reordering contigs to the same order as sequence dictionary file is required for preparing files that can be read by GATK. Individual BAM files must be sorted and indexed before they can be merged.
13. Generation of duplicated DNA sequencing reads is one technical artifact found in many applications. The duplicates can typically arise during PCR, especially when the total number of molecules in the library is small. The detection of duplicated reads is relied on the alignment positions, and works better for paired-end reads. It is because that reads mapped to the same positions at both ends are much more likely to be duplicated than having identical ends by chance. Duplication removal is a necessary step for variant calling, because it removes nonindependence among reads caused by amplification bias. It can also be an optional step for ChIP-seq, if the paired-end approach is used. In RNAseq, however, removing duplicated reads will attenuate the true signals; instead, methods in expression analysis have been developed to account for this other biases [16]. MarkDuplicates work on library level which was determined from the LB field in read group information. Reads mapped to the same positions but with different LB tags are not considered as duplicates.
14. Realignment is a recommended post-processing step for variant calling, as indels at the end of aligned reads often lead to false-positive SNP calls. The indel artifacts arise because mapping tools process one read at a time. Realignment step adjusts the indel position by utilizing all mapped reads so that the overall mismatches are minimized.
15. Quality scores generated from sequencing machine may not be accurate and influenced by several known covariates. Recalibration step is to remove those effects to make quality scores more

a

SAMPLE	NA20322
TOTAL_BASES (Gb)	8.55
PCT_ALGINED_READS	99.3%
PCT_ONTARGET_BASES	40.2%
MEAN_TARGET_COVERAGE	76.8
PCT_TARGET_10X	88.2%
PCT_TARGET_20X	81.2%
PCT_DUP_READS	8.6%

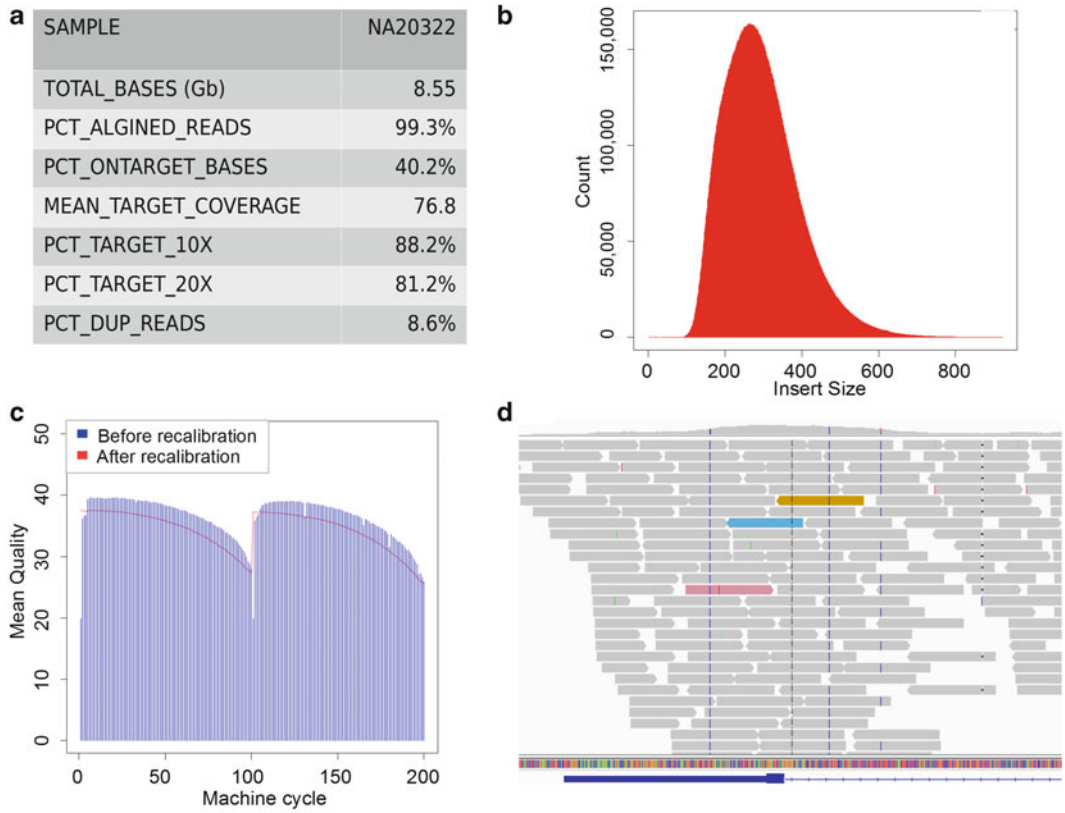

Fig. 2 (**a**) A selected subset of quality control statistics. (**b**) The distribution of insert sizes. (**c**) The mean quality score across machine cycles before and after recalibration. (**d**) Visualization of alignments using Integrated Genome Viewer (IGV)

reflective of true error rate. Special options should be turned on for SOLiD reads:

`--solid_recal_mode REMOVE_REF_BIAS --solid_no-call_strategy PURGE_READ`.

16. Figure 2 shows the expected results for exome sequencing. We consider that the sequencing is complete if more than 80 % of the targets are covered by 20×. It is because not all targets can be adequately covered. The depth of coverage is mainly influenced by the local GC contents, which can be examined by CollectGcBiasMetrics program in picard. The proportion of aligned reads should be typically over 90 %; most unaligned reads have poor base qualities. In our example, the aligned proportion is very high because many low-quality reads have already been filtered out. If an unusually low proportion of mapped reads is observed, then you should be wary of contamination from other species. To test this possibility, FastQ Screen[8] can be used to screen the unmapped reads across

[8]FastQ Screen: http://www.bioinformatics.babraham.ac.uk/projects/fastq_screen/.

several known species and determine the mapping proportions. The unmapped reads can be extracted from alignment and reformatted to FastQ file using the following command:

```
samtoolsview -f 0x0004mapping/NA20322.recal.bam | \
awk '{ print "@"$1"\n"$10"\n+\n"$11;}' > unmapped.fastq
```

17. verifyBamID was designed to detect sample swapping or mixing. The CHIPMIX statistics estimate unusual proportion of non-reference alleles given known genotypes at known sites and corresponding population allele frequencies. In the absence of known SNP genotypes, verifyBamID estimates the proportion of non-reference alleles that is incompatible with that which comes from one individual (the FREEMIX statistics).

Acknowledgments

Xueya Zhou and Suying Bao have contributed equally to this work. This work was funded by grants from the National Basic Research Program of China (No. 2012CB316504) to X.G.Z. and NSFC grants (No. 81271226 to Y.Q.S. and No. 91010016 to X.G.Z.), and from the Research Grants Council of Hong Kong (HKU775208M/HKU777212) and the Research Fund for the Control of Infectious Diseases of Hong Kong (No.11101032) to Y.Q.S.

References

1. Bamshad MJ, Ng SB, Bigham AW, Tabor HK, Emond MJ, Nickerson DA, Shendure J (2011) Exome sequencing as a tool for Mendelian disease gene discovery. Nat Rev Genet. doi:10.1038/nrg3031
2. Ozsolak F, Milos PM (2011) RNA sequencing: advances, challenges and opportunities. Nat Rev Genet 12(2):87–98. doi:10.1038/nrg2934
3. Park PJ (2009) ChIP-seq: advantages and challenges of a maturing technology. Nat Rev Genet 10(10):669–680. doi:10.1038/nrg2641
4. Ning Z, Cox AJ, Mullikin JC (2001) SSAHA: a fast search method for large DNA databases. Genome Res 11(10):1725–1729. doi:10.1101/gr.194201
5. Li H, Homer N (2010) A survey of sequence alignment algorithms for next-generation sequencing. Brief Bioinform 11(5):473–483. doi:10.1093/bib/bbq015
6. Flicek P, Birney E (2009) Sense from sequence reads: methods for alignment and assembly. Nat Methods 6(11 Suppl):S6–S12. doi:10.1038/nmeth.1376
7. Bao S, Jiang R, Kwan W, Wang B, Ma X, Song YQ (2011) Evaluation of next-generation sequencing software in mapping and assembly. J Hum Genet 56(6):406–414. doi:10.1038/jhg.2011.43
8. Holtgrewe M, Emde AK, Weese D, Reinert K (2011) A novel and well-defined benchmarking method for second generation read mapping. BMC Bioinformatics 12:210. doi:10.1186/1471-2105-12-210
9. Trapnell C, Pachter L, Salzberg SL (2009) TopHat: discovering splice junctions with RNA-Seq. Bioinformatics 25(9):1105–1111. doi:10.1093/bioinformatics/btp120
10. Krueger F, Andrews SR (2011) Bismark: a flexible aligner and methylation caller for Bisulfite-Seq applications. Bioinformatics 27 (11):1571–1572. doi:10.1093/bioinformatics/btr167
11. Krawitz P, Rodelsperger C, Jager M, Jostins L, Bauer S, Robinson PN (2010) Microindel

detection in short-read sequence data. Bioinformatics 26(6):722–729. doi:10.1093/bioinformatics/btq027

12. Lunter G, Goodson M (2011) Stampy: a statistical algorithm for sensitive and fast mapping of Illumina sequence reads. Genome Res 21(6):936–939. doi:10.1101/gr.111120.110
13. Hach F, Hormozdiari F, Alkan C, Birol I, Eichler EE, Sahinalp SC (2010) mrsFAST: a cache-oblivious algorithm for short-read mapping. Nat Methods 7(8):576–577. doi:10.1038/nmeth0810-576
14. Cock PJ, Fields CJ, Goto N, Heuer ML, Rice PM (2010) The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. Nucleic Acids Res 38(6):1767–1771. doi:10.1093/nar/gkp1137
15. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R (2009) The Sequence Alignment/Map format and SAMtools. Bioinformatics 25(16):2078–2079. doi:10.1093/bioinformatics/btp352
16. Roberts A, Trapnell C, Donaghey J, Rinn JL, Pachter L (2011) Improving RNA-Seq expression estimates by correcting for fragment bias. Genome Biol 12(3):R22. doi:10.1186/gb-2011-12-3-r22
17. Li H, Durbin R (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics 25(14):1754–1760. doi:10.1093/bioinformatics/btp324
18. Langmead B, Salzberg SL (2012) Fast gapped-read alignment with Bowtie 2. Nat Methods 9(4):357–359. doi:10.1038/nmeth.1923
19. Langmead B, Trapnell C, Pop M, Salzberg S (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. Genome Biol 10(3):R25
20. Homer N, Merriman B, Nelson SF (2009) BFAST: an alignment tool for large scale genome resequencing. PLoS One 4(11):e7767. doi:10.1371/journal.pone.0007767
21. Homer N, Merriman B, Nelson S (2009) Local alignment of two-base encoded DNA sequence. BMC Bioinformatics 10(1):175
22. Hormozdiari F, Hach F, Sahinalp SC, Eichler EE, Alkan C (2011) Sensitive and fast mapping of di-base encoded reads. Bioinformatics 27(14):1915–1921. doi:10.1093/bioinformatics/btr303
23. Rumble SM, Lacroute P, Dalca AV, Fiume M, Sidow A, Brudno M (2009) SHRiMP: accurate mapping of short color-space reads. PLoS Comput Biol 5(5):e1000386. doi:10.1371/journal.pcbi.1000386
24. David M, Dzamba M, Lister D, Ilie L, Brudno M (2011) SHRiMP2: sensitive yet practical short read mapping. Bioinformatics 27(7):1011–1012. doi:10.1093/bioinformatics/btr046
25. Li H, Ruan J, Durbin R (2008) Mapping short DNA sequencing reads and calling variants using mapping quality scores. Genome Res 18(11):1851–1858. doi:10.1101/gr.078212.108