Short-read Alignment with MAQ & BWA

2009-05-01

Wellcome Trust Sanger Institute
Heng Li

After getting short reads...

- Alignment against a known reference sequence:
  - ✓ Resequencing
    - SNPs and short indels
    - structural variations (SVs)
  - ✓ ChIP-seq: binding sites
  - ✓ RNA-seq: expression level and alternative splicing
  - ✓ Survey of methylation pattern
- *De novo* assembly:
  - ✓ unknown reference genome
  - ✓ transcriptome sequencing
  - ✓ local assembly for SVs

Contents

- Focus on:
  - ✓ Alignment to a known reference sequence
  - ✓ Calling SNPs and short indels from alignment
- Will not cover:
  - ✓ Procedures to generate read sequences and qualities
  - ✓ *De novo* assembly

Alignment

Typical input data for alignment

- Illumina/Solexa: 100 million 50+50bp read pairs in a run
- AB/SOLiD: similar in scale and maybe shorter read length
- Roche/454: ~300-500bp reads, 100Mbp a run
  - ✓ Currently a little more expensive in terms of money/base-pair

Difficulties in large-scale alignment

- Speed:
  - ✓ Given a single Illumina run: >200 CPU days using BLAST/BLAT
  - ✓ New algorithms: short-read specific improvements
- Memory:
  - ✓ Suffix array index requires 12GB for human genome
  - ✓ Indexing reads or better in-memory index
- Accuracy:
  - ✓ ~20% of human genome are repetitive to 32bp reads
  - ✓ Effectively using paired-end information

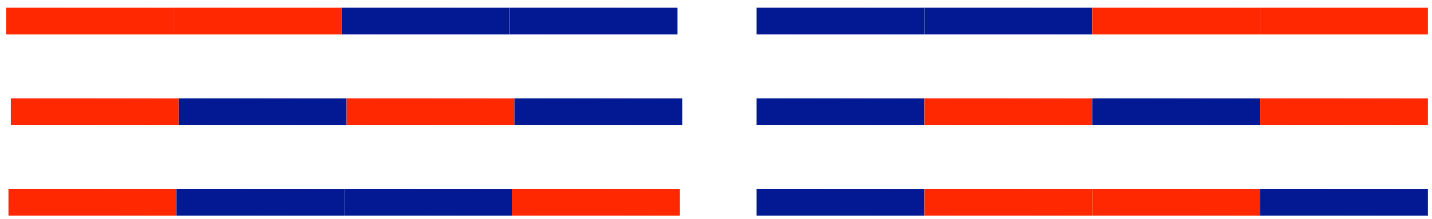
Review of alignment algorithms

- Hashing the reference genome:
  - ✓ Pros: straightforward; easy multi-threading
  - ✓ Cons: large memory
- Hashing read sequences:
  - ✓ Pros: flexible memory footprint
  - ✓ Cons: multi-threading is hard
- Alignment by merge sorting:
  - ✓ Pros: flexible memory
  - ✓ Cons: hard for pairing?
- Indexing genome by BWT:
  - ✓ Pros: fast and relatively small memory footprint
  - ✓ Cons: not applicable to long reads at the moment

MAQ: basic algorithm

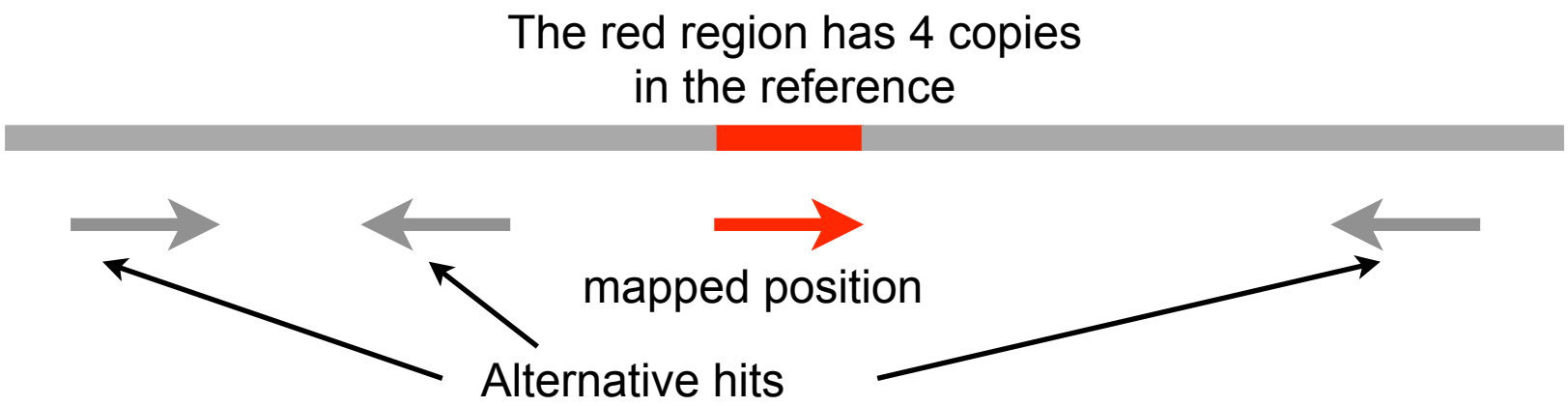
- Index reads and scan the genome.
  - ✓ Avoid aligning too few reads
- 28bp seed; Eland-like indexing
  - ✓ Able to find more mismatches beyond the seed
- Guarantee to find 2-mismatch seed hits

Seed templates:



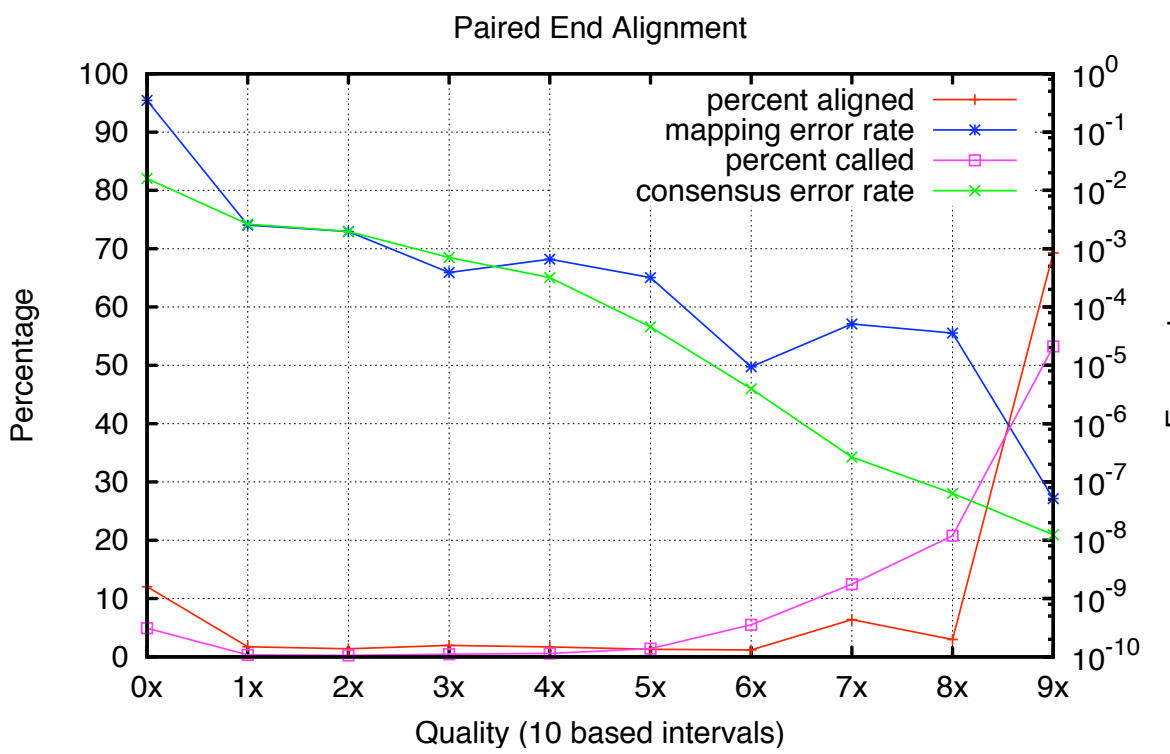
MAQ: random mapping

- Randomly place a read if it has multiple equally best hits
- Advantages:
  - ✓ tell if a read is mapped
  - ✓ tell if a region has reads mapped (avoid holes due to repeats)



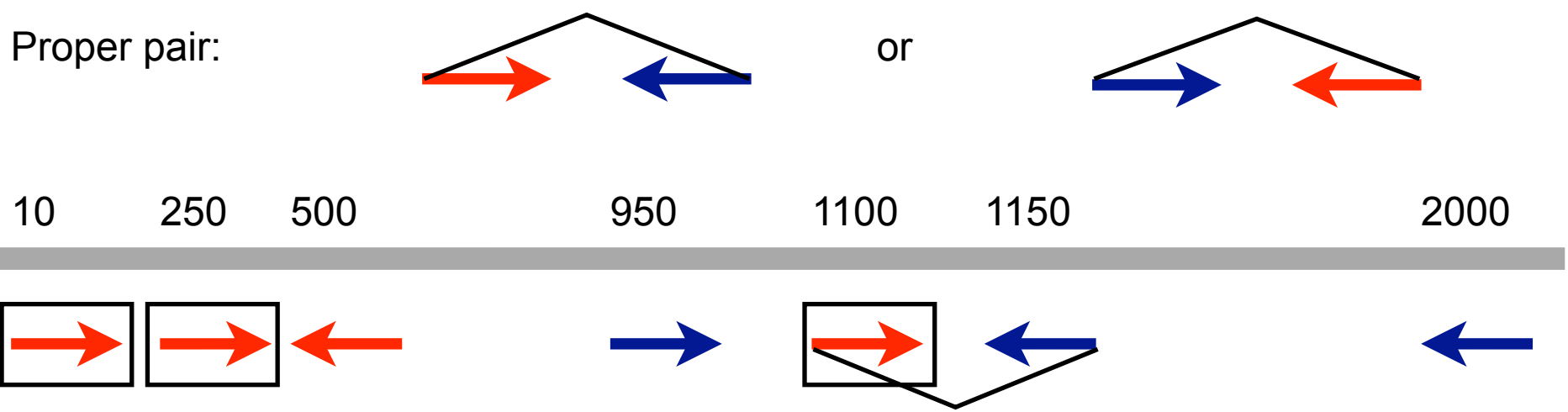
MAQ: mapping quality

- Mapping quality is the phred-scaled probability of the alignment being wrong.
- Discriminate good mappings from bad ones, e.g.:
  - ✓ repetitive reads
  - ✓ top hit is perfect but there are 100 1-mismatch hits
  - ✓ top hit is perfect but the second best hit has one Q5 mismatch
- Proved to be effective for SV detections where wrong alignments dominate.



MAQ: PET mapping

- Hit to a read found on the forward strand: keep the position in a 2-element queue
- Hit to a read found on the reverse strand: check the positions in the queue of its mate



MAQ: other features

- Gapped alignment for PET
- Adapter trimming
- Partially SOLiD mapping support
  - ✓ cannot align the first primer base
- Alignment-based decoding for color reads
  - ✓ correcting color errors after the alignment

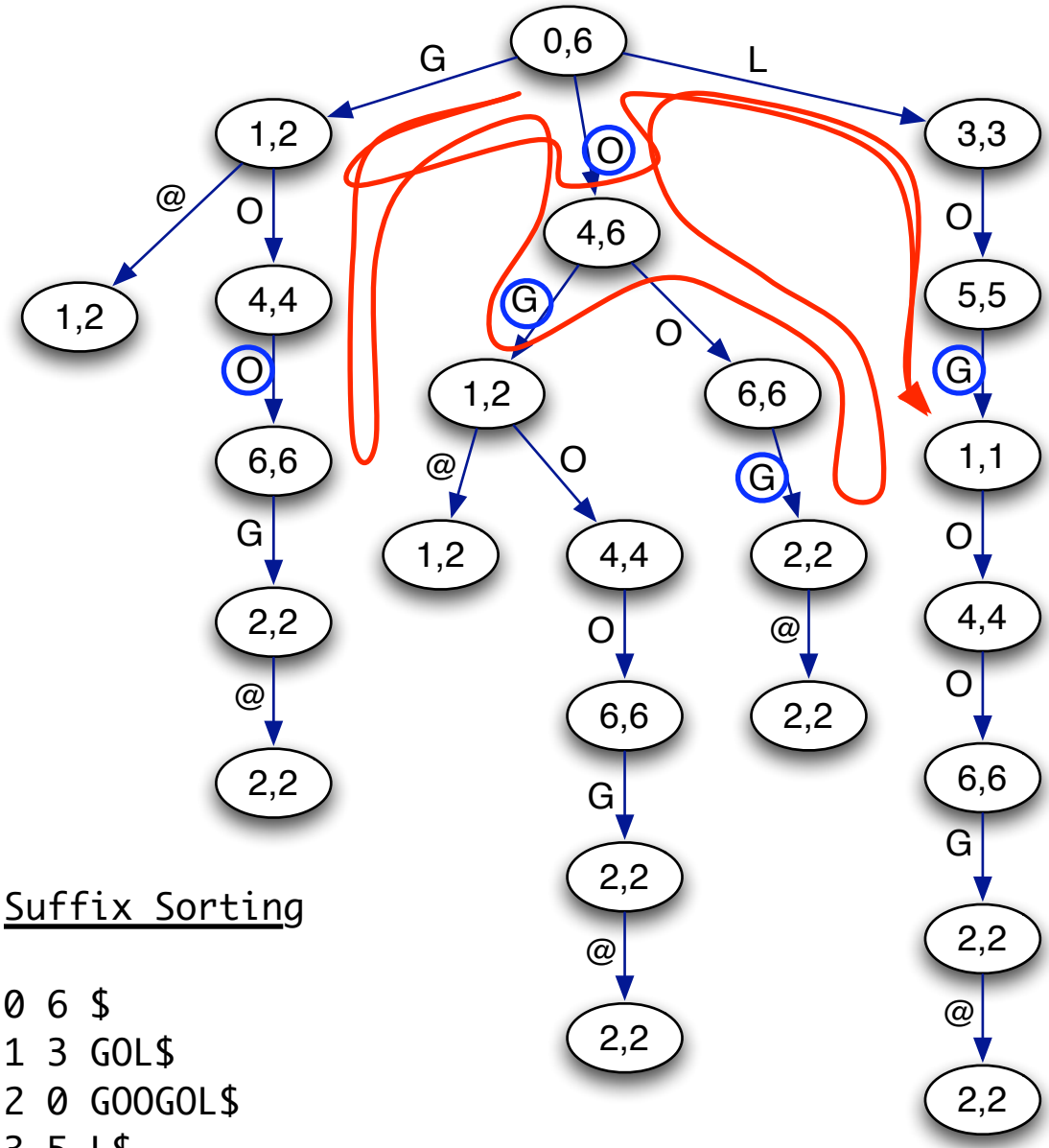
MAQ: problems

- Speed: typically ~100 reads/sec against the human genome
  - ✓ 23 CPU days for a good Illumina run
- No gapped alignment for single-end reads
  - ✓ Not suitable for Helicos reads
  - ✓ Long reads are more likely to contain short indels

BWA

- BWT-based indexing of the reference genome
- ~10X faster than MAQ
- Similar alignment accuracy to MAQ
- Gapped alignment for single-end reads
- SAM output

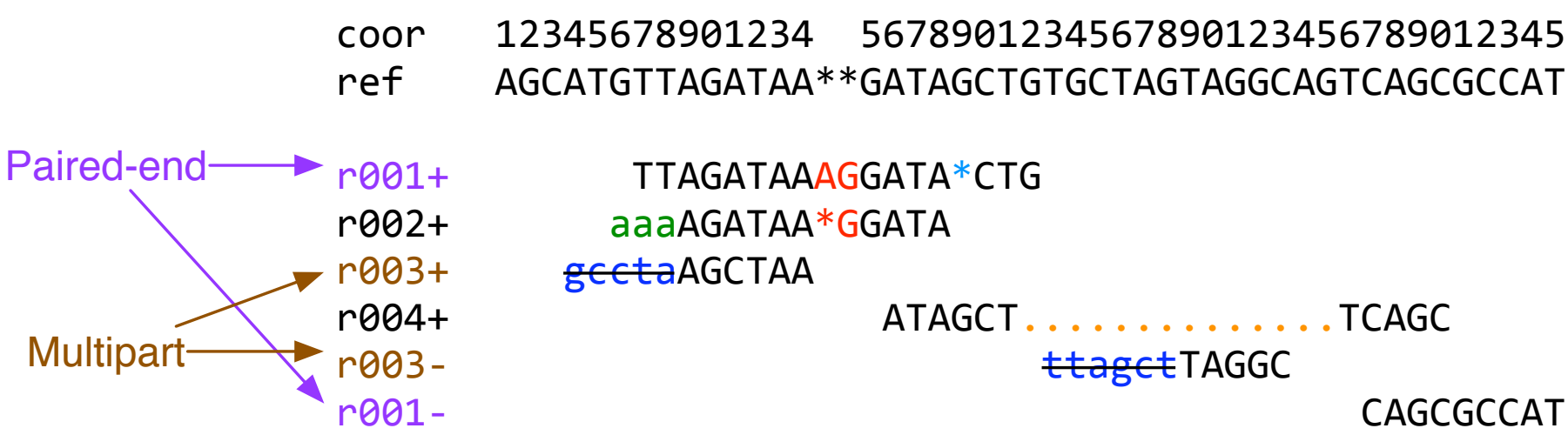
Prefix trie of GOOGOL$



Suffix Sorting

```
0 6 $
1 3 GOL$
2 0 GOOGOL$
3 5 L$
4 6 OGOL$
5 4 OL$
6 1 OOGOL$
```

Task: search LOL allowing one mismatch



Ins & padding
Soft clipping
Splicing
Hard clipping

```
@SQ SN:ref LN:45
r001 163 ref  7 30 8M2I4M1D3M = 37  39 TTAGATAAAGGATACTA *
r002   0 ref  9 30 3S6M1P1I4M * 0   0 AAAAGATAAGGATA    *
r003   0 ref  9 30 5H6M       * 0   0 AGCTAA    *   NM:i:1
r004   0 ref 16 30 6M14N5M    * 0   0 ATAGCTTCAGC       *
r003  16 ref 29 30 6H5M       * 0   0 TAGGC     *   NM:i:0
r001  83 ref 37 30 9M         = 7 -39 CAGCGCCAT         *
```

```
ref  7 T 1 .    | ref 12 T 3 ...        | ref 17 T 3 ...
ref  8 T 1 .    | ref 13 A 3 ...        | ref 18 A 3 .-1G..
ref  9 A 3 ...  | ref 14 A 2 .+2AG.+1G  | ref 19 G 2 *.
ref 10 G 3 ...  | ref 15 G 2 ..         | ref 20 C 2 ..
ref 11 A 3 ..C  | ref 16 A 3 ...        | ...
```

Alternative FREE aligners

- cross_match, SSAHA2 and Mosaik:
  - ✓ Pros: 454 and capillary reads; local alignment
  - ✓ Cons: SSAHA2 is a little slow for short reads
- NovoAlign:
  - ✓ Pros: most accurate to date
  - ✓ Cons: relatively large memory; a little slow
- Bowtie and SOAP2:
  - ✓ Pros: fastest (also BWT based)
  - ✓ Cons: less accurate than MAQ; Bowtie for ungapped alignment only
- Tophat: RNA-seq

Recommended alignment procedures

- Long reads: BLAT/SSAHA2/cross_match/Mosaik
- Long reference (e.g. human genome), short reads:
  - ✓ BWA for initial mapping (for speed)
  - ✓ NovoAlign for unmapped/unpaired reads (for accuracy, in particular for detecting structural variations)
  - ✓ Local *de novo* assembly with PE reads for structural variations
- Short reference (e.g. bacterial genome), short reads:
  - ✓ Speed/memory is less critical
  - ✓ *De novo* assembly + cross_match contigs (to find variants in highly variable regions, but require deep coverage)

Postprocessing of Alignment and Variant Calling

Reference-based assembly

- Task: collect read bases at each reference position.
- The MAQ/SAMtools way (small memory footprint):
  - ✓ Do alignment in a batch of a few million reads
  - ✓ Sort alignment based on chromosomal positions
  - ✓ Merge alignments from multiple batches
  - ✓ Generate assembly on a stream

Consensus calling by MAQ/ SAMtools

- Bayesian consensus caller to calculate the probability of the call being wrong
- Explicitly consider:
  - ✓ base quality and mapping quality
  - ✓ dependence between errors
- Indel caller:
  - ✓ MAQ: simply count the number of reads supporting indels
  - ✓ SAMtools: redo alignment locally around indels (higher power)

Alternative SNP callers

- Implementing SNP callers is complicated by alignment formats.
- Much fewer SNP callers and most are aligner specific:
  - ✓ Slider (for slider)
  - ✓ SOAPsnp (for soap and soap2)
  - ✓ GigaBase (for Mosaik)
  - ✓ SHORE (for vmatch and genomemapper)

Software for other applications

- ChIP-seq: FindPeaks (actively maintained, open source)
- SV detection: BreakDancer (work for MAQ alignment, http://genome.wustl.edu/tools/cancer-genomics/)

Alignment viewers

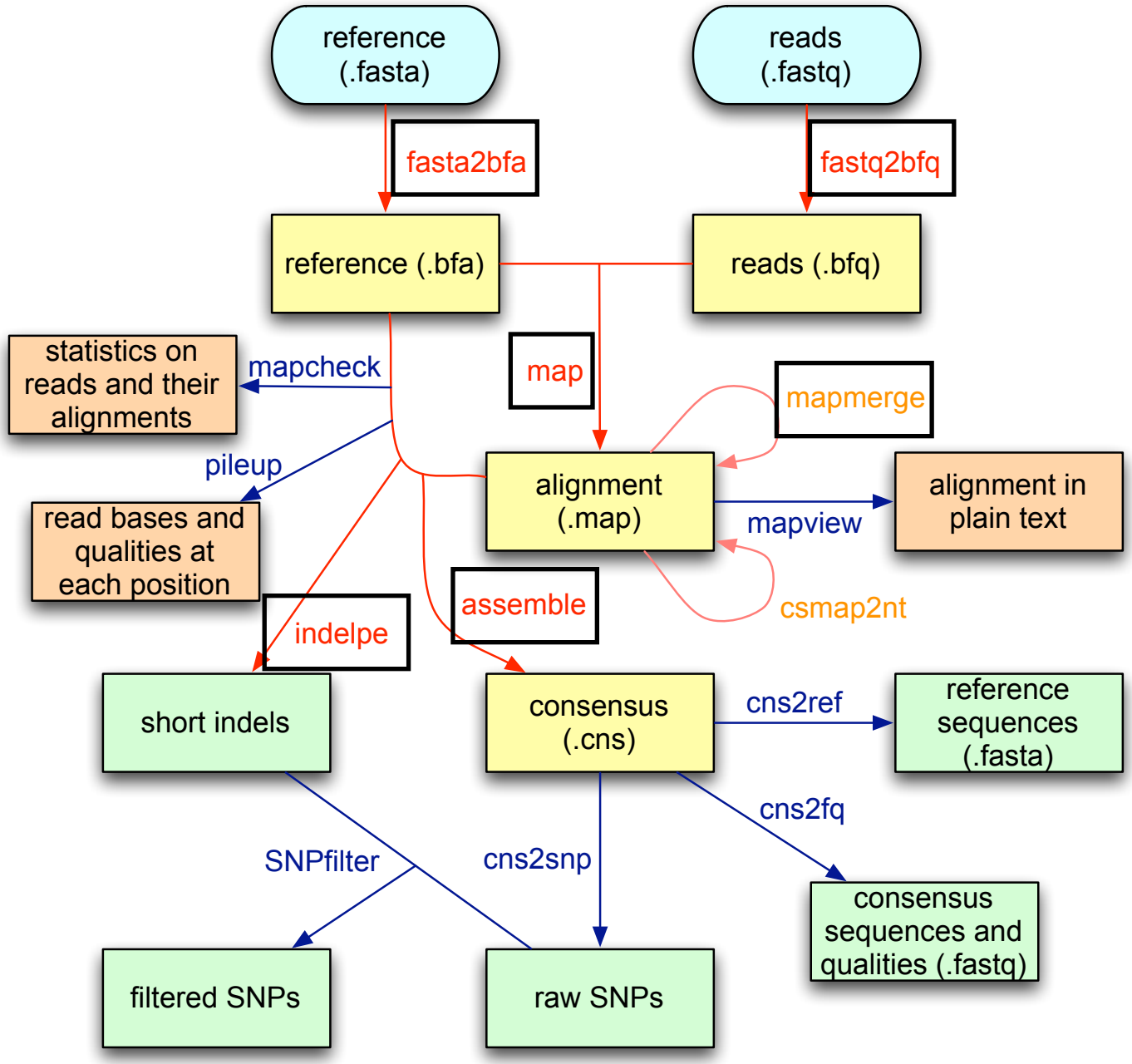
- SAMtools
  - ✓ http://samtools.sourceforge.net
- MAQview
  - ✓ http://maq.sourceforge.net
- MapView
  - ✓ http://evolution.sysu.edu.cn/mapview/
- IGV
  - ✓ http://www.broad.mit.edu/igv-beta/

Using MAQ/BWA/ SAMtools

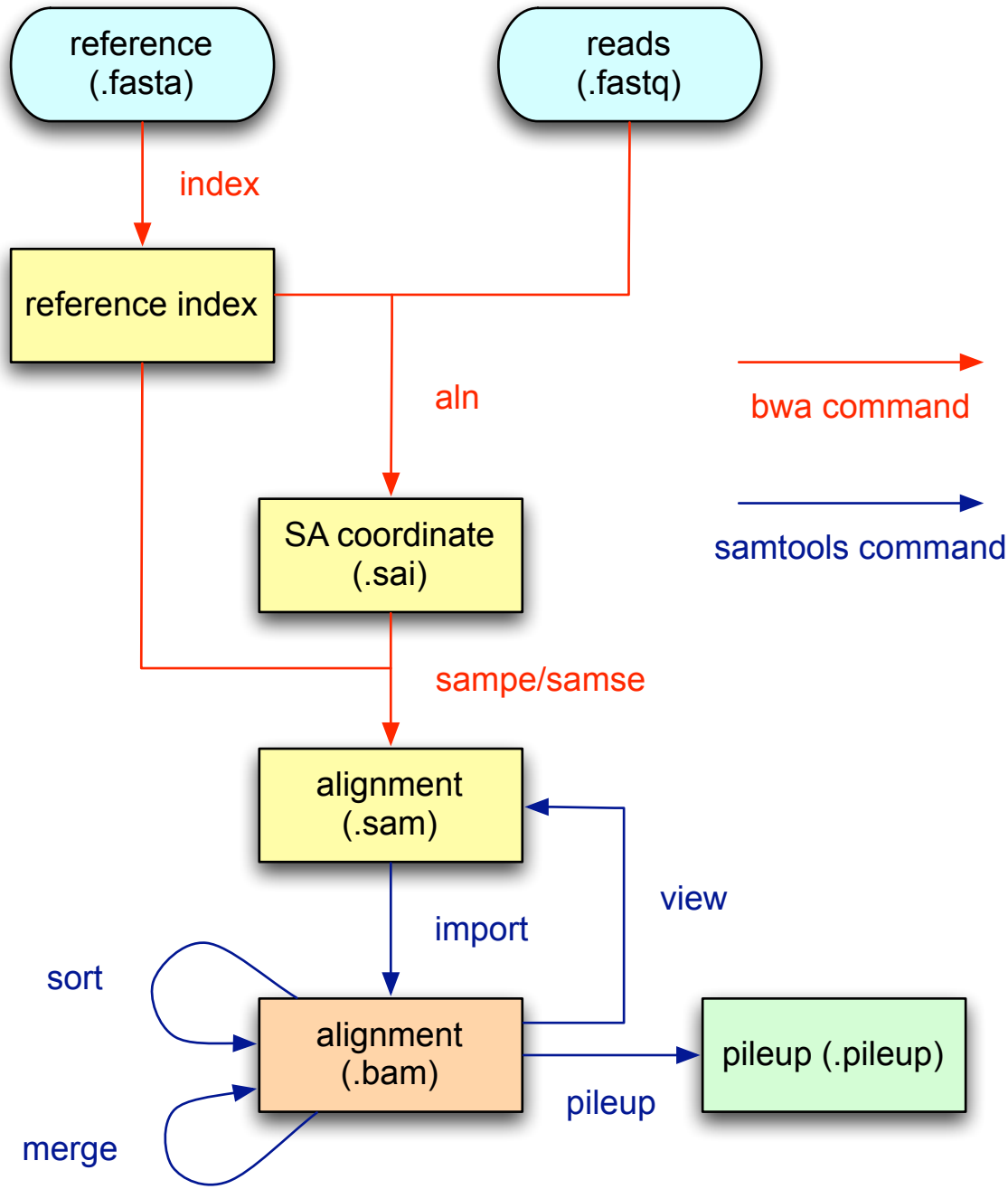
The two pipelines

- MAQ:
  - ✓ Publication proved
  - ✓ More matured
- BWA+SAMtools:
  - ✓ 10X faster for human alignment with similar alignment accuracy
  - ✓ Gapped alignment for single-end reads
  - ✓ Improved short indel caller
  - ✓ Bleeding-edge

MAQ Work Flow



BWA/SAMtools Work Flow



MAQ pitfalls: alignment (I)

- Too many reads in a batch
  - ✓ MAQ's memory is linear in the #reads in the alignment
  - ✓ Too many reads make MAQ use too much memory
- Too few reads in a batch
  - ✓ MAQ's speed is similar given 100 reads and 100,000 reads
  - ✓ Recommendation: 2 million reads or 1 million pairs in a batch
- Assertion failure for paired end reads
  - ✓ In PET mapping, i-th read in the first file and i-th in the second file constitute a read pair
  - ✓ Two reads in a pair must have identical read names OR only differ at the tailing "/[12]": read001/1 and read001/2

MAQ pitfalls: alignment (II)

- Wrong '-e' option for long reads:
  - ✓ -e controls the tolerance of mismatches across the full read
  - ✓ -n controls the max-mismatches in the 28bp seed
- Improper '-a' option:
  - ✓ -a sets the maximum insert size
  - ✓ Excessively small -a leads to more wrong alignments
  - ✓ Recommendation: it is safer to use larger -a, although the resultant mapping quality would be a little conservative.
- Highly inaccurate base qualities:
  - ✓ Lead to inaccurate mapping quality (though not highly)
  - ✓ Recommendation: calibrate base qualities

Qualities

- base quality
  - ✓ Solexa quality
  - ✓ Fastq quality
- mapping quality
- consensus quality
- SNP quality
- see also:
  - ✓ http://en.wikipedia.org/wiki/FASTQ_format
  - ✓ http://maq.sourceforge.net/qual.shtml
  - ✓ http://maq.sourceforge.net/pooled.shtml

BWA pitfalls

- Forget to apply seeding with ‘bwa aln -l 32’:
  - ✓ Seeding greatly accelerates BWA at a marginal cost of accuracy.
- Use BWA for 454 reads:
  - ✓ BWA works but has high false positive rate.

Pitfalls in variant calling

- Unfiltered SNPs:
  - ✓ The statistical model does not consider all the artifacts.
  - ✓ Run ‘maq.pl SNPfilter’ is highly recommended.
- Pooled samples:
  - ✓ Should use ‘maq assemble -N’ (See also online documentations)
- SNPs with excessive coverage:
  - ✓ Most likely due to segmental duplications in the sample

Write your own SNP caller

- Extended pileup format

```
2  2044247 T  20  .$,$,.,.,.G.,,,,.,....  ;<<<<<<<)<=<<<;<<9<;
2  2044248 T  19  ,.,.,.,.,,,,,,....^cA   <<<<<<)<=<<<<<<<<<
2  2044249 G  20  ,$.$,.,.,.,,,,,....C^eC        <4<0<;.<=<<.7<:1<<4+
2  2044250 T  21  ,.,.,.,.,,,,,....AA^jA^kA^hA     <<<<3<=<<<<<<1<;<<<=;
2  2044251 T  21  ,.,.,.,.,,,,,.,.,.,..   <<<<;<=<<<<<<1<461<=<
2  2044252 C  24  ,.,.,.,.,,,,,....AAAAA^~A^~A^~.  <9<;7<=<<<<<<2<<<<<=<++<
2  2044253 A  26  ,$.$,.,.,.,.,.,..+5CATAG..+5CATAG.+5CATAGGGGGGGG.^~.^~.  <<<<9<;<<<<<<<<<8-:=<++<<;
2  2044254 A  24  CCC.,,,,.,.,.,.,.,.,.,.,.        ++;<;<<<<<<<<<;;<=<<<<<<
2  2044255 C  25  A$A$A.,,,,,,,A..,,,,,,,,,.^~,   ++7<=<<<<<<;<<;6;=<<<<<<<
```


01 2044211 2044221 2044231 2044241 2044251 2044261 2044271 2044281 2044291 2044301 2044311 2044321
AGTTCTAAATATAGAAATTGAAACAGCTGTGTTTAGTGCCTTTGTTCA*****ACCCCCTTGCAACAACCTTGAGAACCCCAGGGAATTTGTCAATGTCAGGGAAGGAGCATTTTGTCAGTTACCAAATG
W.MR
CATAG
CATAG
CATAG
aca,ag*****
ca,ag*****
a,ag*****
A.AG*****
A.AG*****
AG*****
AG*****
CA
CATAG

```
# simulate reads
./wgsim -N 200000 -1 40 -2 40 ssuisP17_cut.fasta r1.fq r2.fq > var.snp

# MAQ easyrun
./maq.pl easyrun -pa 700 ssuisP17_cut.fasta r1.fq r2.fq
./maqindex -ic easyrun/consensus.cns easyrun/all.map

# build BWA index
./bwa index -a is ssuisP17_cut.fasta
# generate suffix array coordinate
./bwa aln ssuisP17_cut.fasta r1.fq > r1.sai
./bwa aln ssuisP17_cut.fasta r2.fq > r2.sai
# pairing
./bwa sampe ssuisP17_cut.fasta r1.sai r2.sai r1.fq r2.fq > pe.bwa.sam
# multiple single-end hits
./bwa samse -n 100 ssuisP17_cut.fasta r1.sai r1.fq > r1.txt
./bwa samse -n 100 ssuisP17_cut.fasta r2.sai r2.fq > r2.txt

# index fasta
./samtools faidx ssuisP17_cut.fasta
# import BWA alignment
./samtools import ssuisP17_cut.fasta.fai pe.bwa.sam pe.bwa.bam
# sort the alignment
./samtools sort pe.bwa.bam pe.srt
# index the alignment
./samtools index pe.srt.bam
# consensus calling
./samtools pileup -cf ssuisP17_cut.fasta pe.srt.bam | gzip > pe.srt.pileup.gz

# ./maqview -c easyrun/consensus.cns easyrun/all.map
# ./samtools tview pe.srt.bam ssuisP17_cut.fasta
```

Thank You!