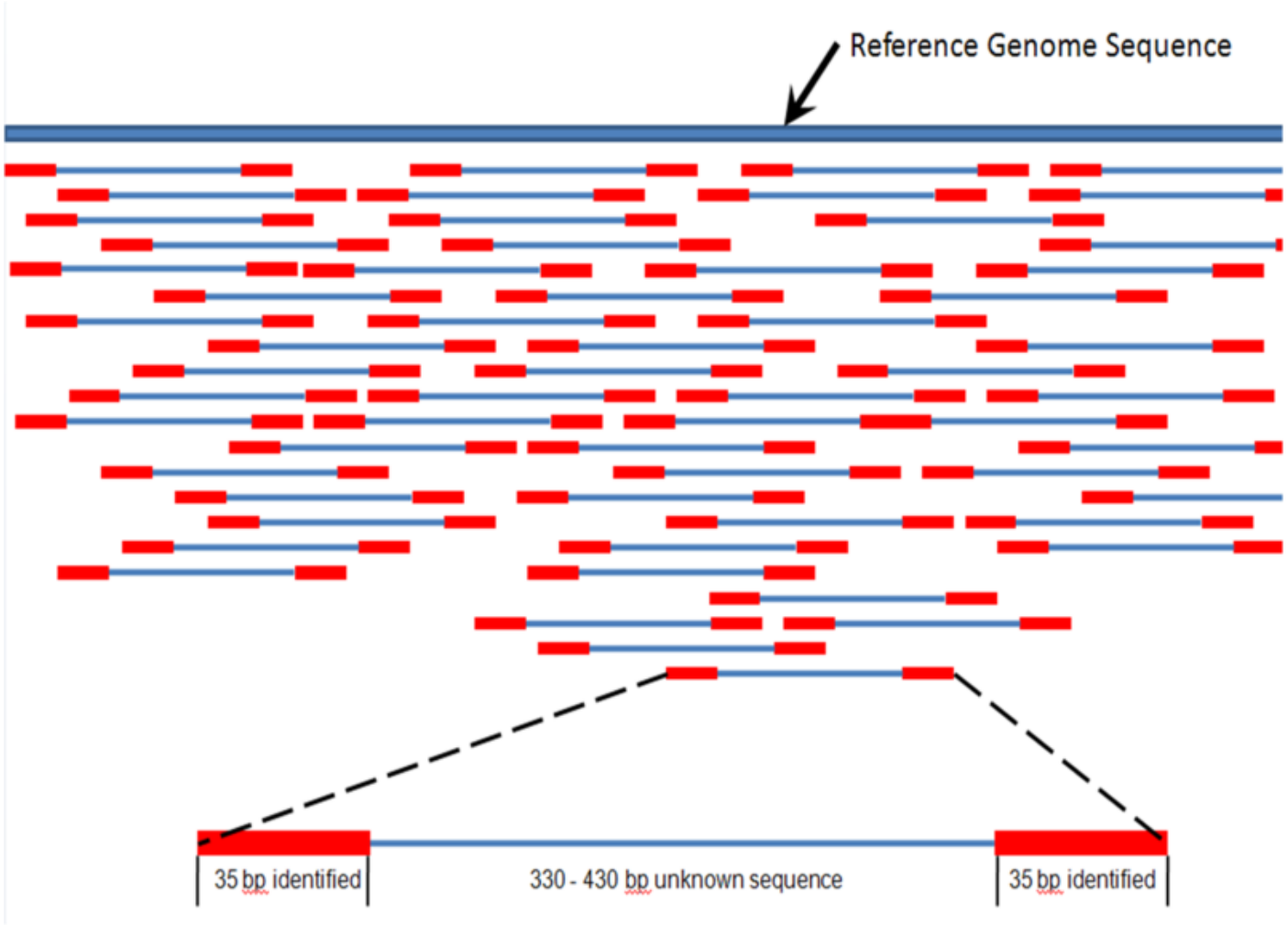
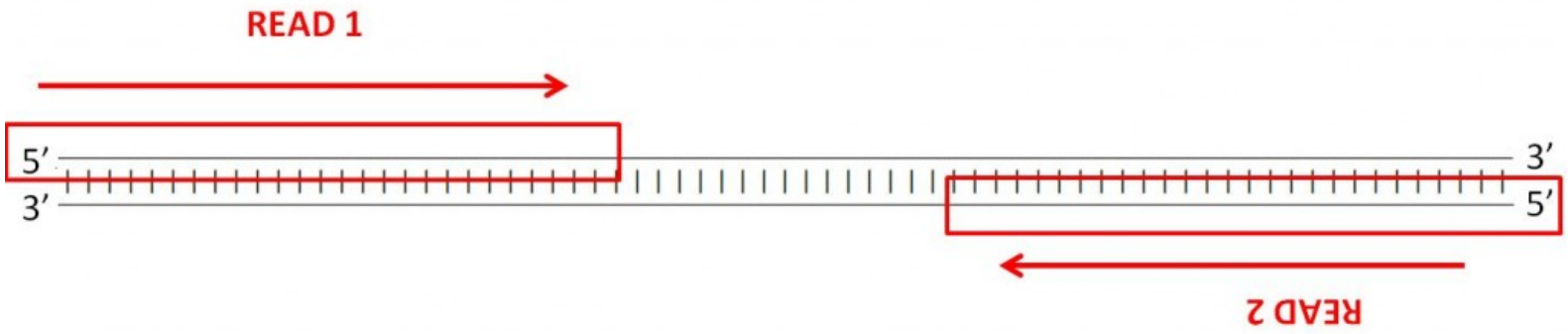
Mapping short reads

Locate reads in ref genome



http://en.wikipedia.org/wiki/File:Mapping_Reads.png



http://www.cureffi.org/2012/12/19/forward-and-reverse-reads-in-paired-end-sequencing/

SNP calling – which variants are "real"?

http://www.kenkraaijeveld.nl/genomics/bioinformatics/

Note: long v short

- Mapping *long* reads is a different problem from mapping short reads.

- This is for two reasons, both of them pragmatic/practical:
  - The volume of data has traditionally been much less: 1m 454 reads vs 200m Illumina
  - Long reads are much more likely to have insertions or deletions in them

Long reads: BLAST vs 'blat'

- On Tues, I told you about the 11-mer heuristic that BLAST uses for DNA matches.
  - BLAST requires that a query sequence contains the same 11-mer as a database sequence before it attempts further alignment.
  - Any given 11-mer occurs only once in 2m sequences, so this filters out many database sequences quickly.
  - You can also store the list of all possible 11-mers in memory easily (~2mb), making it possible to keep track of everything quickly.
- 'blat' does the same thing as BLAST, but is faster because it uses longer k-mers.

How *alignment* works, and why indels are the devil

There are many alignment strategies, but most work like this:

```
GCGGAGatggac        GCGGAGatggac
||||||......  =>    ||||||x.....
GCGGAGgcggac        GCGGAGgcggac
```

At each base, try extending alignment; is total score still above threshold?

How *alignment* works, and why indels are the devil

There are many alignment strategies, but most work like this:

```
GCGGAGatggac        GCGGAGatggac
||||||......  =>    ||||||xx....
GCGGAGgcggac        GCGGAGgcggac
```

Each mismatch *costs*.

How *alignment* works, and why indels are the devil

Insertions/deletions introduce *lots* more ambiguity:

```
GCGGAGagaccaacc      GCGGAGag-accaacc
||||||          =>  ||||||
GCGGAGggaaccacc      GCGGAGggaacc-acc

GCGGAGagaccaacc      GCGGAGaga-ccaacc
||||||          =>  ||||||
GCGGAGggaaccacc      GCGGAGggaacca-cc
```

Mapping short reads, again

- What's hard about mapping
- Three mapping programs
- Decisions to be made
- Color space issues

Mapping, defined

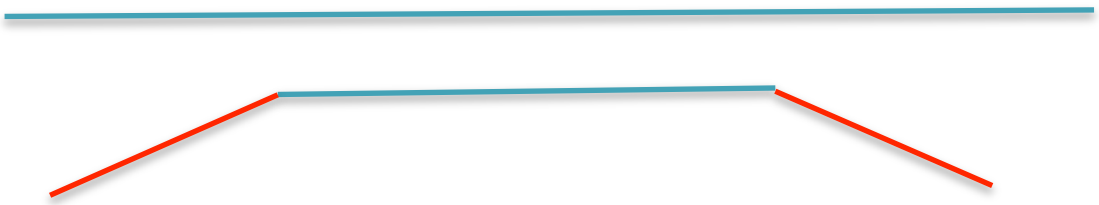
- Exhibit A: 20m+ reads from genome/ transcriptome.
- Exhibit B: related genome/transcriptome, aka "the reference"

- Goal: assign all reads to location(s) within reference.

- Req'd for resequencing, ChIP-seq, and mRNAseq

Want *global*, not *local*, alignment

- You do not want matches *within* the read, like BLAST would produce.



- Do not use BLAST!

Mapping is "pleasantly parallel"

- Goal is to assign each individual read to location(s) within the genome.

- So, you can map each read *separately*.

What makes mapping challenging?

- Volume of data
- Garbage reads
- Errors in reads, and quality scores
- Repeat elements and multicopy sequence
- SNPs/SNVs
- Indels
- Splicing (transcriptome)

Volume of data

- Size of reference genome is not a problem: you can load essentially all genomes into memory (~3 gb).

- However, doing *any* complicated process 20m times is generally going to require optimization!

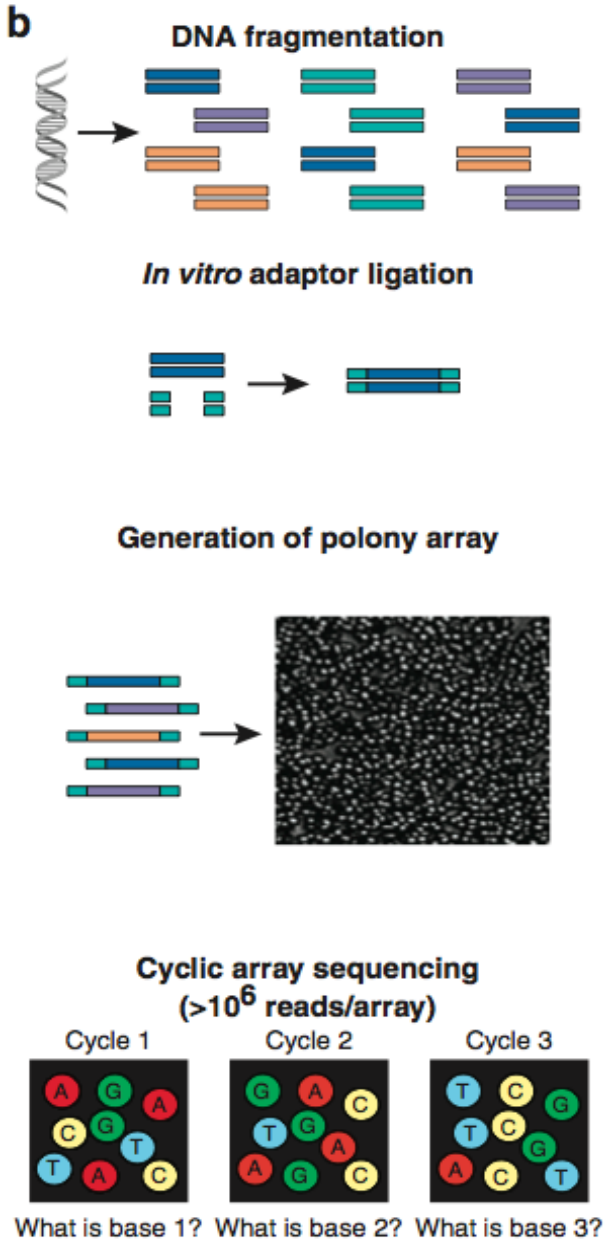
Garbage reads

Overlapping polonies result in mixed signals.

These reads will not map to anything!

Used to be ~40% of data.

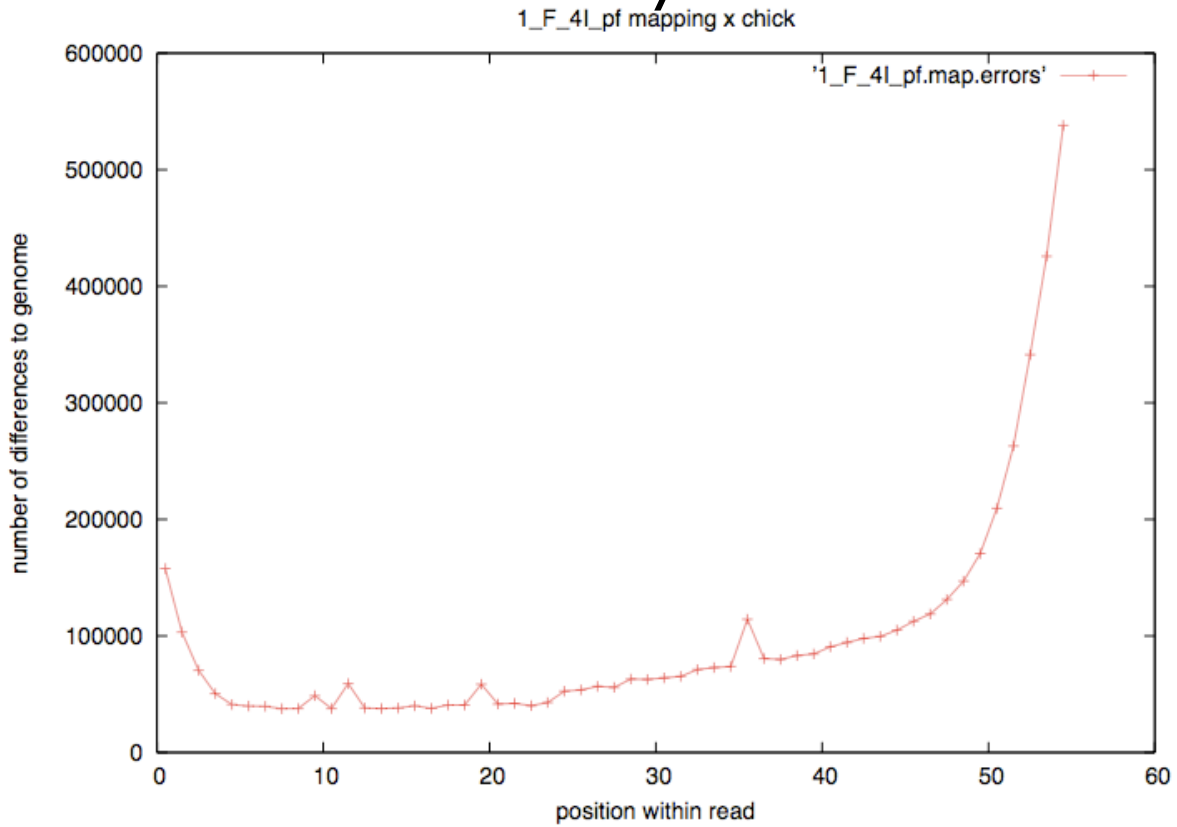
Increasingly, filtered out by sequencing software.



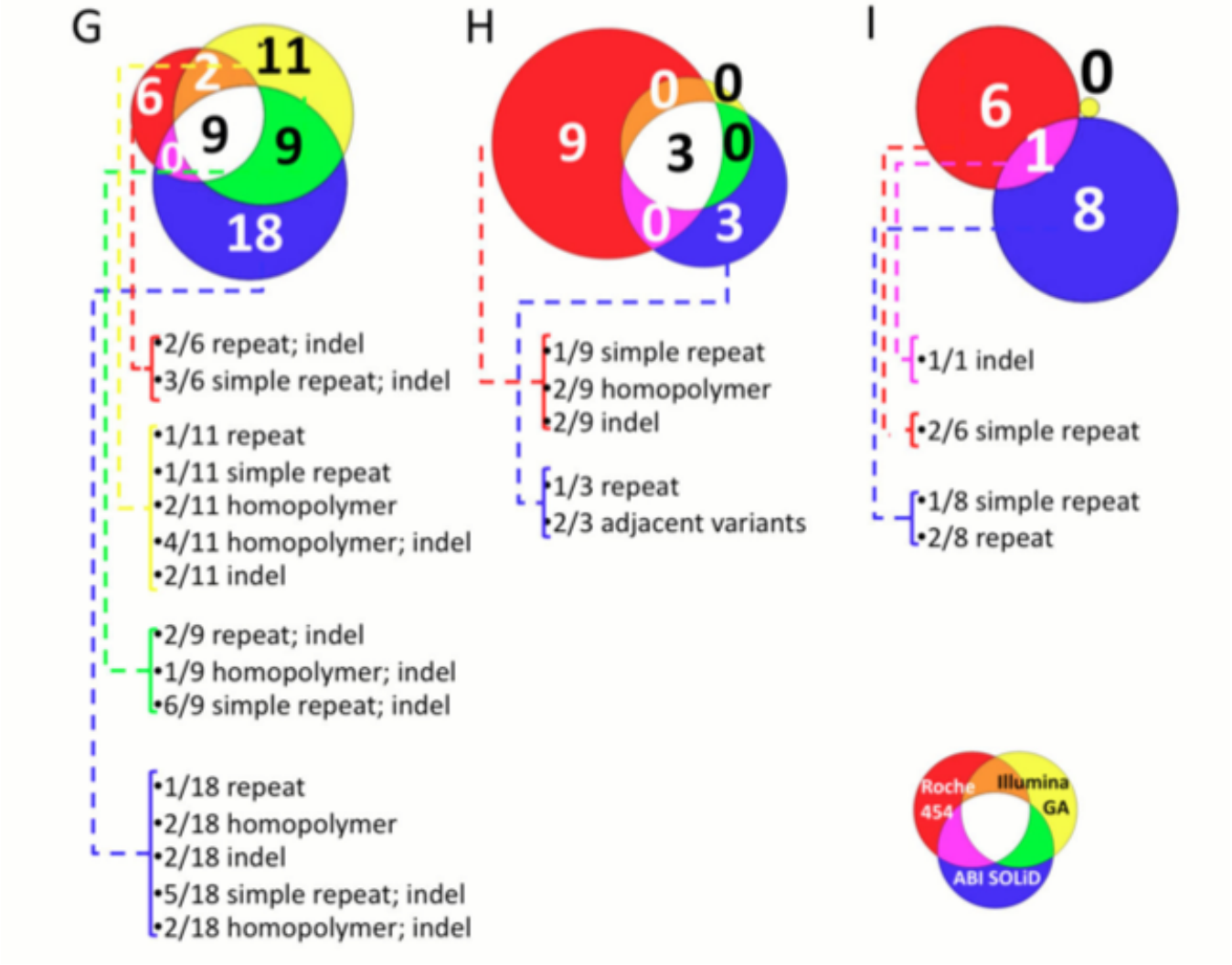
Shendure and Ji, Nat Biotech, 2008

Errors in reads

When mapping, a mismatch is not necessarily "real".



Technology-specific bias



Harismendy et al., Genome Biol. 2009, pmid: 19327155

Errors in reads

- Quality scores are based on Sanger sequencing-style quality scores: per base.
- But 454 is subject to different biases than Illumina, and the biases are not necessarily base-by-base (think: homopolymer runs)

Repeat/multi-copy elements

- Multi-copy sequence makes it impossible to map all reads uniquely.

- Repeats are particularly bad, because there are (a) lots of them and (b) they vary in sequence. They therefore may "attract" reads depending on what optimizations/heuristics you use.

SNP/SNVs

- Genuine mismatches between reference and sequence *do* exist, of course.
  - Polymorphism
  - Diploidy
  - Population studies
- You want to map these reads!
- Fast heuristic approaches exist, based on fuzzy matching.
- However, they are still biased towards mapping exact matches.
  - This can be a problem for allelotyping and population studies.
  - Likit will discuss next week.

Indels

- Remember, they are the devil…?
- Complicate mapping heuristics
- Complicate *statistics*

Indels: ambiguity & decisions…

```
TGACGATATGGCGATGGACTGGACG
|x|||||||||||x||x|x||||||
TcACGATATGGCGgTGaA-TGGACG

TGACGATATGGCGATGGACTGGACG
|x|||||||||||x|x||x||||||
TcACGATATGGCGgT-GAaTGGACG
```

Splice sites

- If you are mapping transcriptome reads to the genome, your reference sequence is different from your source sequence!
- This is a problem if you don't have a really good annotation.

- Main technique: try to map across splice sites, build new exon models.
- Another technique: assembly.

Two specific mapping programs

- Bowtie
- BWA

Both open source.

BWA is widely used now, so we'll use that for examples.

(There are many more, too.)

Bowtie I

- Not indel-capable.

- Designed for:
  - Many reads have one good, valid alignment
  - Many reads are high quality
  - Small number of alignments/read

a.k.a. "sweet spot" :)

BWA

- Uses similar strategy to Bowtie, but does gapped alignment.
- Newest, hottest tool.

Decisions to be made by you

- How many mismatches to allow?
  - Vary depending on biology & completeness of reference genome
- Report how many matches?
  - Are you interested in multiple matches?
- Require best match, or first/any that fit criteria?
  - It can be much faster to find *first* match that fits your criteria.

All of these decisions affect your results and how you treat your data.

Mapping best done on *entire reference*

- May be tempted to optimize by doing mapping to one chr, etc. "just to see what happens"
- Don't.
- Simple reason: if you allow mismatches, then many of your reads will match erroneously to what's present.

Look at your mapping

Just like statistics ☺

This is a lot of what we'll be doing today.

Two considerations in mapping

- Building an index
  - Prepares your "reference"
  - (Not really a big deal for single microbial genomes)

Indexing – e.g. BLAST

BLASTN filters sequences for exact matches between "words" of length 11:

```
GAGGGTATGACGATATGGCGATGGAC
||x|||||x|||||||||||x|x||x
GAcGGTATcACGATATGGCGgT-Gag
```

What the 'formatdb' command does (see Tuesday's first tutorial) is *build an index* ("index") sequences by their 11-base word content – a "reverse index" of sorts.

Indexing – e.g. BLAST

What the 'formatdb' command does (see Tuesday's first tutorial) is *build an index* ("index") sequences by their 11-base word content – a "reverse index" of sorts.

Since this index only needs to be built once for each reference, it can be slower to build – what matters to most people is *mapping speed.*

All short-read mappers have an indexing step.

Speed of indexing & mapping.

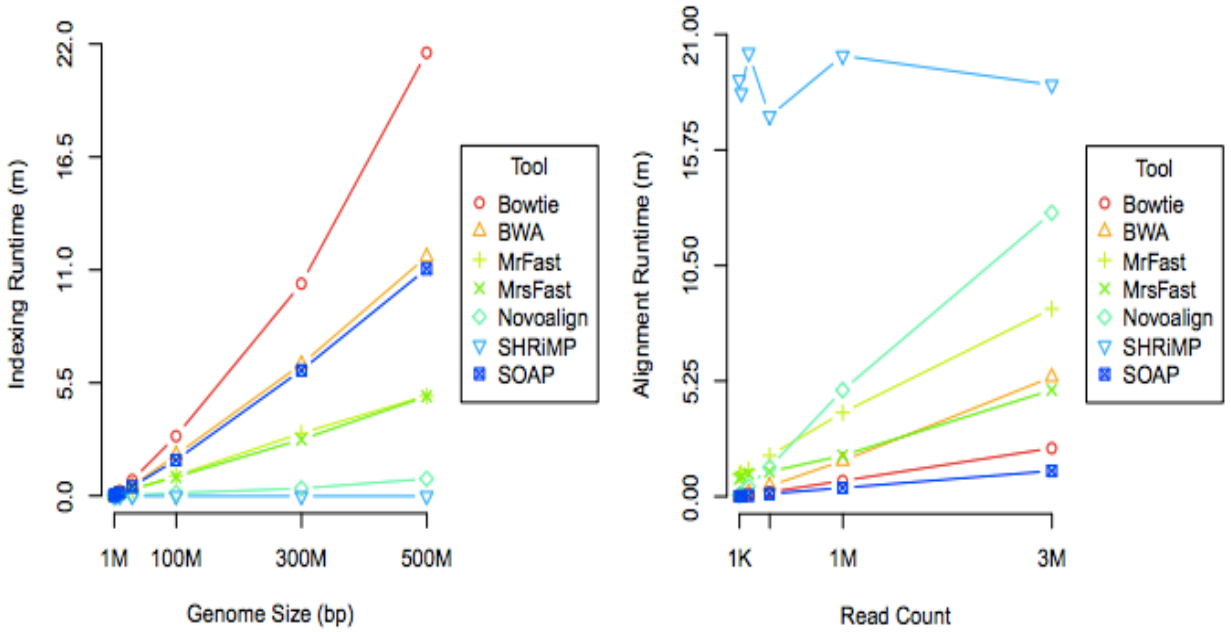


Fig. 5: Runtime measurements: (a) shows indexing time vs. genome size, (b) shows alignment time vs. read count on a 500Mbp genome.

Fig 5 of Ruffalo et al. PMID 21856737, Bioinformatics 2011.

Simulations => understanding mappers

Table 1. Read mapping errors for single (SE) and paired end (PE) reads from random (simulated) and real transcriptomes

Organism	Num Trans	Error	TP (d)	FP (d)	TP (u)	FP (u)	TP (m)	FP (m)
Random (SE)	5000	1%	92%	0%	92%	0%	92%	0%
Mouse (SE)	5000	1%	87%	5%	81%	0%	92%	12%
Random (PE)	5000	1%	85%	0%	85%	0%	85%	0%
Mouse (PE)	5000	1%	81%	4%	77%	0%	85%	9%

Mapping parameters are default (d), unique (u), and multimap (m). True positives are reads that were successfully mapped to their originating transcript. False positives are reads that were mapped to other transcripts (even if the read was an exact match to the alternate transcript).

Mappers will ignore some fraction of reads due to errors.

Pyrkosz et al., unpub.; http://arxiv.org/abs/1303.2411

Does choice of mapper matter?
Not in our experience.

Table 3. Comparison of Three Common Mapping Programs on the Same Chicken Data Sets

Organism	Num Trans	Bowtie TP (d)	FP (d)	BWA TP (d)	FP (d)	SOAP2 TP (d)	FP (d)
Chicken	100%	78%	22%	78%	20%	78%	22%
Chicken	90%	72%	21%	72%	20%	72%	21%
Chicken	80%	65%	22%	65%	21%	65%	22%
Chicken	70%	58%	22%	58%	21%	58%	22%
Chicken	60%	51%	20%	50%	19%	51%	20%
Chicken	50%	44%	19%	44%	18%	44%	19%
Chicken	40%	36%	16%	37%	16%	36%	17%
Chicken	30%	27%	13%	27%	13%	27%	12%
Chicken	20%	19%	11%	19%	11%	19%	11%
Chicken	10%	9%	5%	9%	6%	9%	5%

Comparison of Bowtie, BWA, and SOAP2 mapping programs on the same simulated reads for error-free chicken read sets (triplicate and averaged) with decreasing completeness of the reference transcriptome, showing equivalent results.

Reference completeness/quality matters more!

Pyrkosz et al., unpub.; http://arxiv.org/abs/1303.2411

Misc points

- Transcriptomes and bacterial genomes have very few repeats…
- …but for transcriptomes, you need to think about shared exons.

- For genotyping/association studies/ASE, you may not care about indels too much.

- Variant calling is less sensitive to coverage than assembly (20x vs 100x)

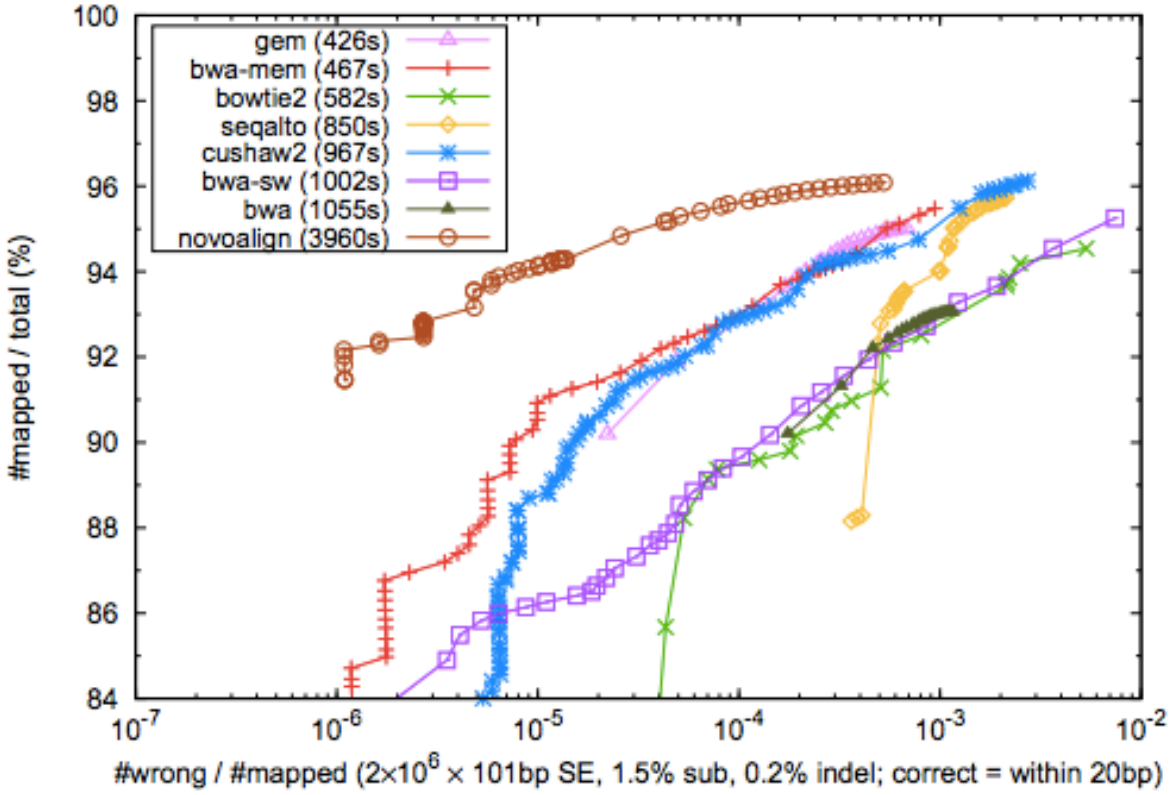
Using quality scores?

- Bowtie uses quality scores; bwa does not.

- This means that bowtie can align some things in FASTQ that cannot be aligned in FASTA.

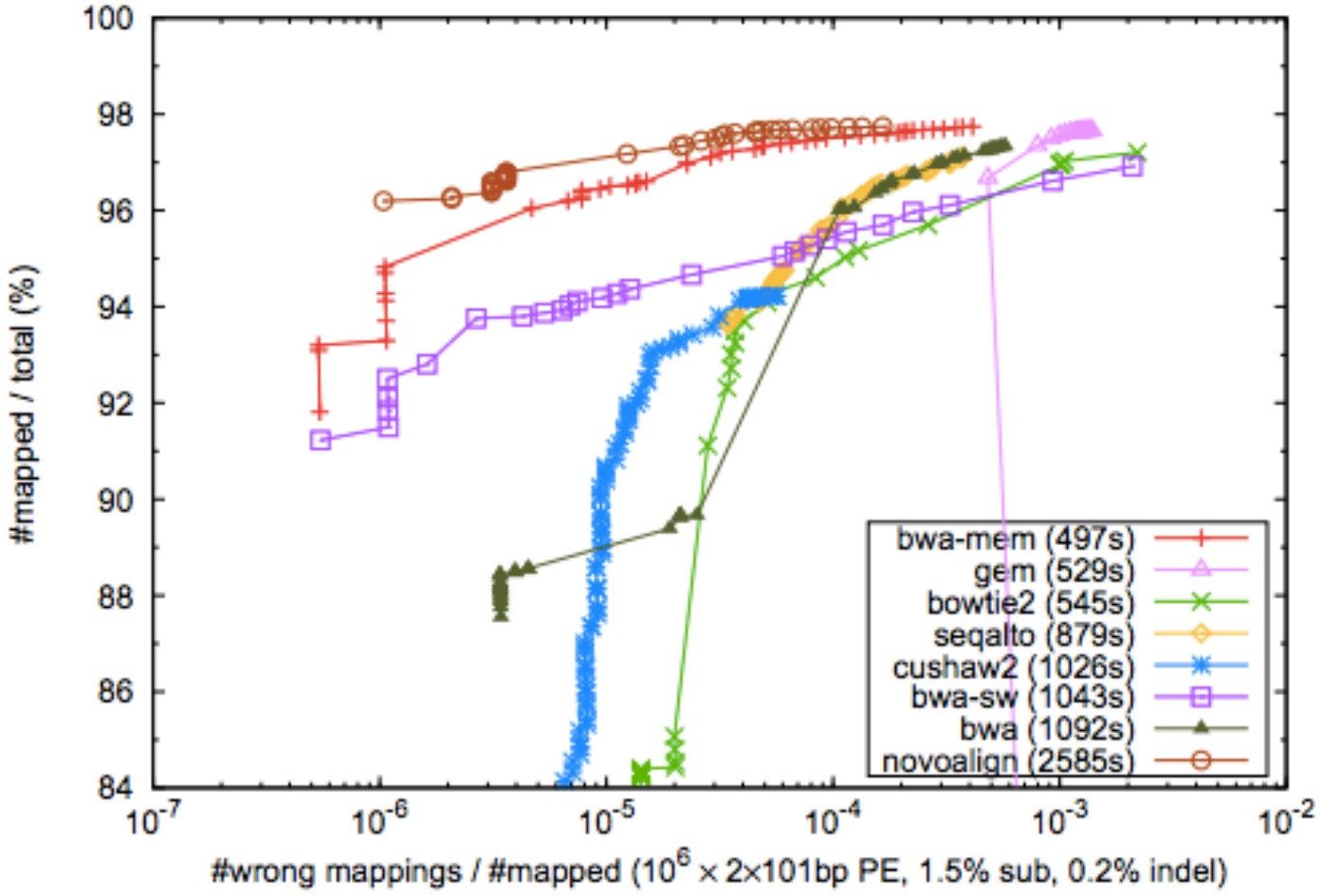
  See: http://www.homolog.us/blogs/blog/2012/02/28/bowtie-alignment-with-and-without-quality-score/

Comparative performance/SE



Heng Li, BWA-MEM: http://arxiv.org/pdf/1303.3997v2.pdf

Comparative performance/PE



Heng Li, BWA-MEM: http://arxiv.org/pdf/1303.3997v2.pdf