Workshop Schedule

- http://omgn.org/workshop/
  - Schedule has links to introductory presentations and the FungiDB workshops

	Monday 24th	Tuesday 25th	Wednesday 26th
AM Session 1	Introductory Presentations	FungiDB Exercises 2	FungiDB Exercises 9
AM Session 2	Introductory Presentations	FungiDB Exercises 4 ,5	FungiDB Exercises 10, 11
PM Session 3	FungiDB Exercise 1	FungiDB Exercises 6, 7	FungiDB Exercises 13
PM Session 4	FungiDB Exercises 1, 3 Discussion	FungiDB Exercises 8 Discussion	FungiDB Exercises 14 Discussion

FungiDB Exercises

- http://fungidb.org/fungidb/
  - Part of EuPathDB, uses same software
- FungiDB is less mature and not as much data has been loaded; not all application are accessible
- When possible Oomycete workshop exercises use oomycete genomes
  - For exercises that use data or applications not yet ready in Fungi DB we will do a similar exercise on a EupathDB organism.

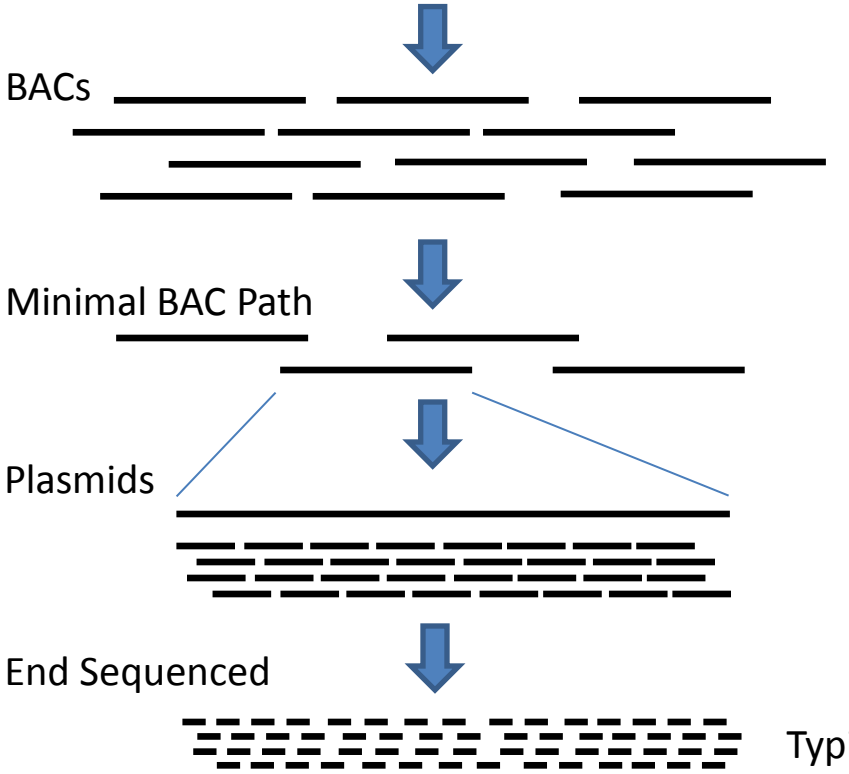
Gene and Genome Sequencing

Brent Kronmiller
Center for Genome Research and Biocomputing
Oregon State University

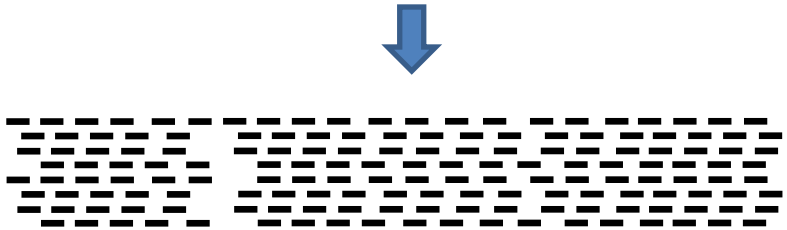
Genome Sequencing

- We still can't sequence a chromosome from start to end.
- The genome is broken into fragments, these are sequenced and assembled to reconstruct the genome.

Hierarchical Shotgun Sequencing



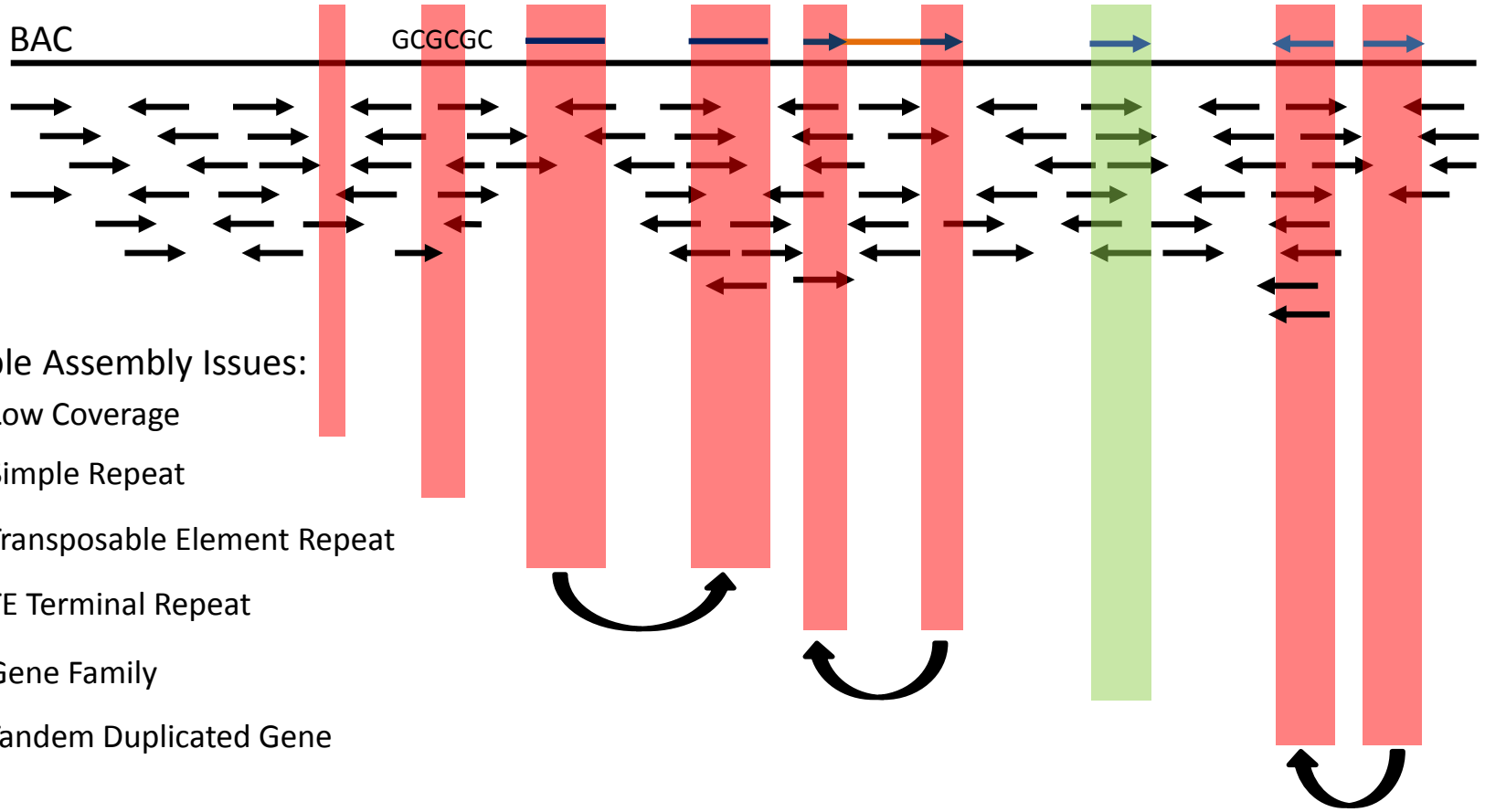
Whole Genome Shotgun Sequencing



- Sequenced at much higher depth
- But locations of sequences are random across whole genome, some areas will be sparse

Typically sequenced at 6-8x coverage before finishing

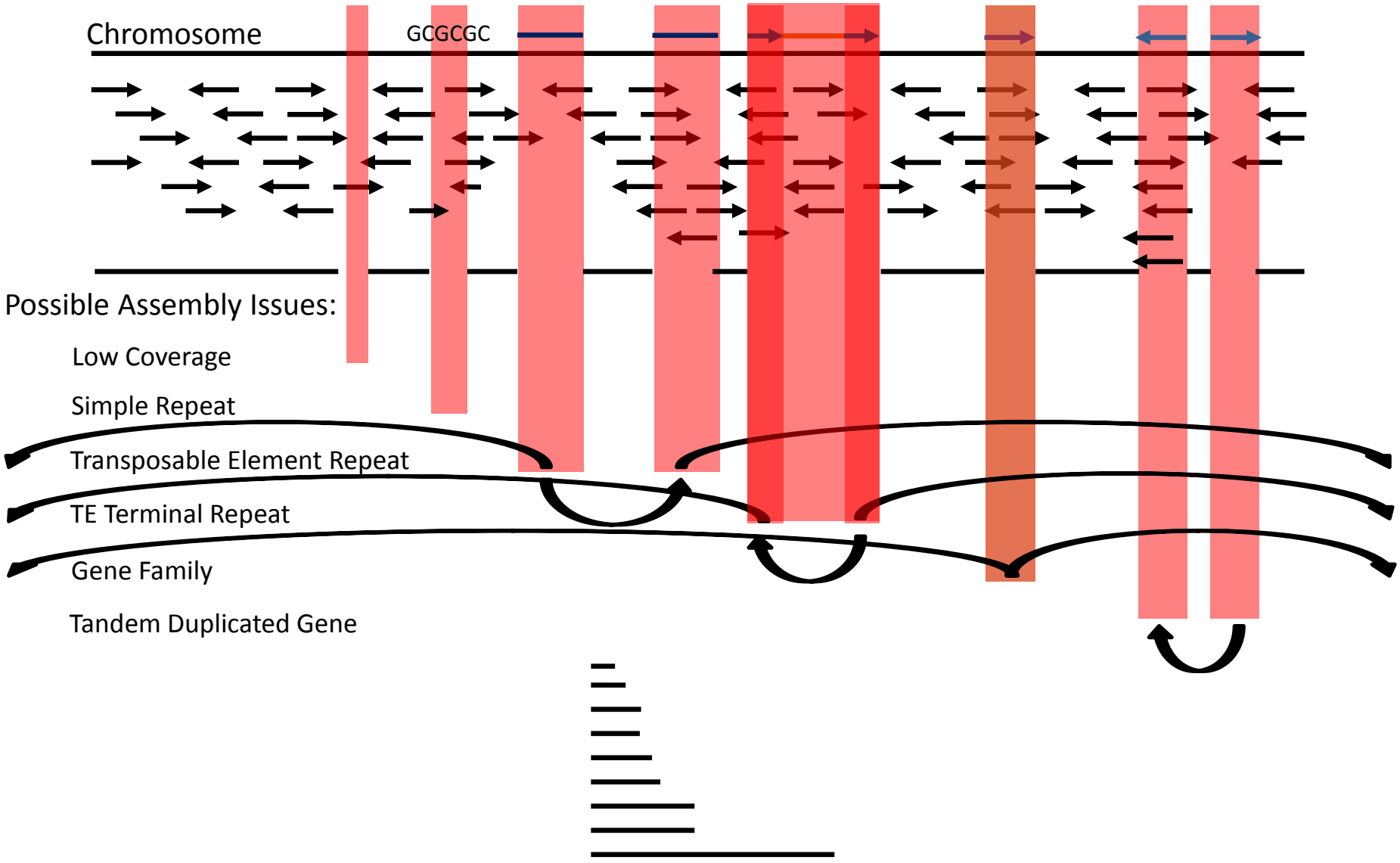
Hierarchical Assembly Results



Constrained mate paired ends can alleviate some of these assembly issues.

- The few gaps can be closed with finishing techniques
- Relative location on genome is known

Whole Genome Shotgun Results



Solutions for WGS

- What can we do about these issues?
  - Paired end sequences
  - Mate pair sequencing (long range)
  - Longer sequences
  - Hybrid sequence types (sanger + NGS)
  - Long range libraries to span issues

Or, only target the genes

- Gene region enhancement for WGS
  - Methyl filtration, HiCot
- Transcriptome sequencing
  - As a hybrid assembly
- RNAseq
  - With reference
  - *de novo*

Sequence Quality Scores

- Each base of a sequence is assigned a quality score.
- Quality is (usually) used by the assembler or aligner to determine the validity of the overlap and the consensus quality.
- Phred log scale:

Phred Score	Chance the base was called incorrectly
Q10	1 in 10
Q20	1 in 100
Q30	1 in 1,000
Q40	1 in 10,000

- Q40 is the max score for a sequence base. Depending on the calling software bases in a sequence can be influenced by surrounding bases and show a score higher than Q40.

FASTQ Files

- FASTQ is the standard flat file sequence read format
- Extension of FASTA, but with quality scores for each base

FASTA

```
>DB775P1:230:D1U9KACXX:5:1101:1474:1950 1:N:0:CGATGT
CAGGNTGGTAGTCAAAGGATTGTTTTTTCCTGTAAGCATCTCATCAGGTGAATAAATGACTTCTCCAGTATCTGG
```

FASTQ

```
@DB775P1:230:D1U9KACXX:5:1101:1474:1950 1:N:0:CGATGT
CAGGNTGGTAGTCAAAGGATTGTTTTTTCCTGTAAGCATCTCATCAGGTGAATAAATGACTTCTCCAGTATCTGA
+
@@@F#2ADADFFHIJIIIGIGIGIJIJJJJJJCEGIJIJGHGIJIIIJ?FGIGGIJGCHGCHGIJJFJIJIGEHH
```

- Quality scores characters can be translated to Phred scores by looking up the Ascii value (Decimal value – 33)
  - Illumina has had 4 version of FASTQ quality scores – be careful that you're using the right version.

equals Q2
2 equals Q17
F equals Q37

Alignments vs Assemblies

- Alignment: align your sequences to a reference genome
  - Quicker, each sequence is compared to the reference sequence
- Assembly: *de novo* reconstruction of genome from sequences
  - Slower, each sequence is compared to each other

- Applications:

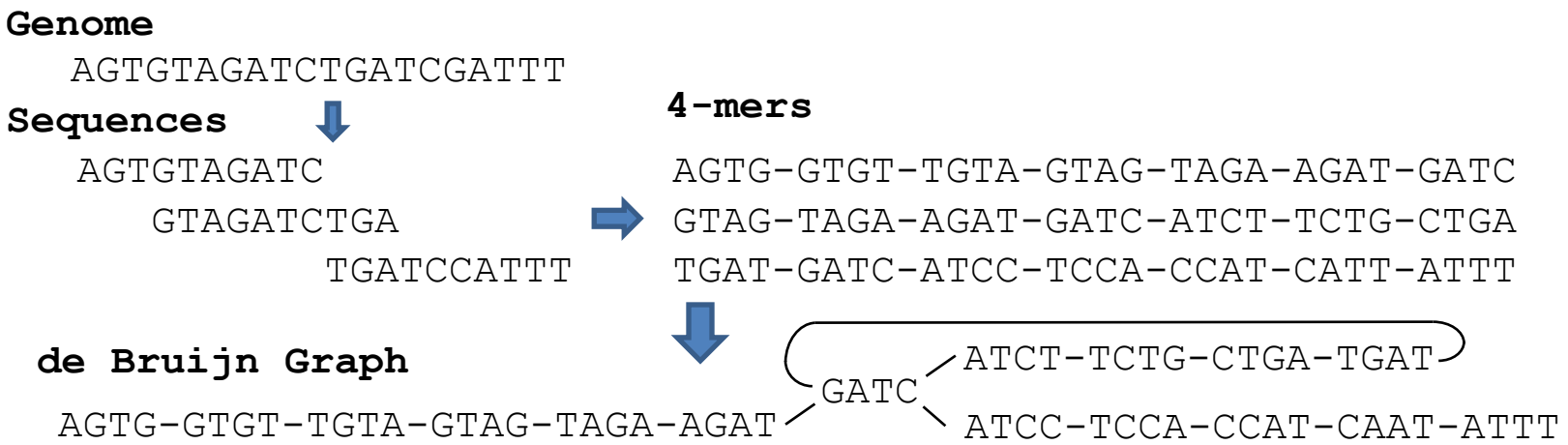
Alignment	Assembly
SNP Identification	Genome Sequencing
RNAseq	Transcriptome Seq
ChIPseq	
Re-Sequence	

Alignment Programs

- Alignment programs use one of two algorithms;
  - Hash table
  - Burrows Wheeler Transformation (BWT)
- Hash table is a data structure in programming
  - Quick lookup of exact matching sequence
  - Sorting is not necessary
  - Programs that use Hash
    - MAQ, ELAND, SHRiMP, SOAP
- BWT
  - Faster than Hash aligners
  - Reference genome is pre-processed into a quickly searchable sorted index, subsequent assemblies will not need to reindex
  - Programs that use BWT
    - BWA, Bowtie, SOAP

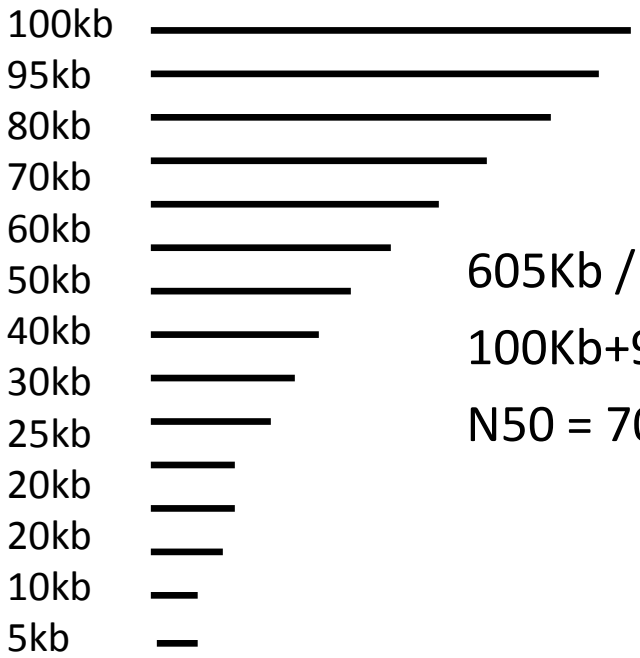
Assembly Programs

- Hierarchical assemblers can use an overlap match based assemblers
  - Phrap, arachne, etc
- A WGS assembly, expecially with NGS will have too many sequences
- Many short read assemblers use de Bruijn graph algorithm
  - ABySS, Velvet, ALLPATHS, SOAPdenovo
  - Uses fixed-length K-mer substrings
  - Assembler doesn't store sequences, just counts of K-mers



N50 for Assembly Assessment

- To calculate N50 for an assembly:
  - Order all contigs produced in the assembly by size
  - Calculate total length of all contigs
  - Find the contig where 50% of the total length of all contigs are found in that contigs of that size or greater, e.g.



605Kb / 2 = 302.5Kb

100Kb+95Kb+80Kb+70Kb = 345Kb

N50 = 70Kb

  - N50 does not assess quality, either from sequence quality or correctness of sequence overlaps

Assembly/Alignment File Formats

- Assembly: output is a FASTA file
  - Multiple FASTA of the contigs – contiguous sequence assemblies
  - Multiple FASTA of the scaffolds – contigs joined when their order and orientation is known by spanning sequences from paired-end or mate pair sequences
- Alignment: SAM/BAM has become the standardized output format
    - SAM: Sequence Alignment/Map format
    - BAM: Binary SAM
  - Binary format allows for quicker retrieval and indexing of information
  - For each sequence read give information on where it aligns and how it matches

SAM/BAM Format

- Header section (optional)
  - Form of @RECORD TAG:VALUE TAG:VALUE …
  - RECORD and TAG are 2-letter codes
  - 5 RECORD and 25 TAG categories
    some are required, some optional
- Alignment section
  - One line per sequence in assembly
  - 11 required fields

@HD – header of the header
VN – SAM version
SO – sorting order

```
@HD VN:1.3 SO:coordinate
@SQ SN:ref LN:45
```

@SQ – Reference
SN – Ref name
LN – Ref length

```
r001 163 ref  7  30 8M2I4M1D3M = 37 39 TTAGATAAAGGATACTG *
r002   0 ref  9  30 3S6M1P1I4M  *  0  0 AAAAGATAAGGATA    *
r003   0 ref  9  30 5H6M        *  0  0 AGCTAA           *
r004   0 ref 16 30 6M14N5M     *  0  0 ATAGCTTCAGC      *
```

  - 37 possible optional tags: TAG:TYPE:VALUE format found after field 11

1	Seq name	7	Ref Mate
2	Bit Flag	8	Position Mate
3	Ref name	9	Insert Length
4	Position	10	Sequence (RC)
5	Map quality	11	Phred Quality
6	CIGAR string		

- Notice header line begins with '@', same as FASTQ header. If your assembler doesn't remove the '@' from the sequence name the alignment section will get confused for the header section

Bitwise Flag

- A numerical code to give you the status of the read in the assembly
  - 163 = 1+2+20+40+100
    - Has pair
    - Has alignment
    - Pair is reversed
    - First of the pair to align
    - 2nd best alignment
  - 12 = 2+10
    - Has alignment
    - Reverse-complemented
  - 83 = 1+2+80
    - Has pair
    - Has alignment
    - Second of the pair to align

Bit	Description
0x1	template having multiple segments in sequencing
0x2	each segment properly aligned according to the aligner
0x4	segment unmapped
0x8	next segment in the template unmapped
0x10	SEQ being reverse complemented
0x20	SEQ of the next segment in the template being reversed
0x40	the first segment in the template
0x80	the last segment in the template
0x100	secondary alignment
0x200	not passing quality controls
0x400	PCR or optical duplicate

CIGAR String

- A compact view of the sequence in the assembly
  - 8M2I4M1D3M
    - 8 bases match
    - 2 bases inserted in sequence relative to ref
    - 4 bases match
    - 1 base deleted in sequence relative to ref
    - 3 bases match

```
ref …ATGTTAGATAA**GATAGCTGTGC…
seq     TTAGATAAAGGATA*CTG
```

  - 6M14N5M
    - 6 bases match
    - 14 bases skipped
    - 5 bases match

```
ref …TGATAGCTGTGCTAGTAGGCAGTCAGCGCC…
seq    ATAGCT..............TCAGC
```

M	Alignment Match
I	Insertion to the reference
D	Deletion from the reference
N	Skipped sequence (intron in RNAseq)
S	Soft clipped
H	Hard clipped
P	Padded
=	Sequence Match
X	Sequence Mismatch