NGS Read Processing

STAMPS 2016

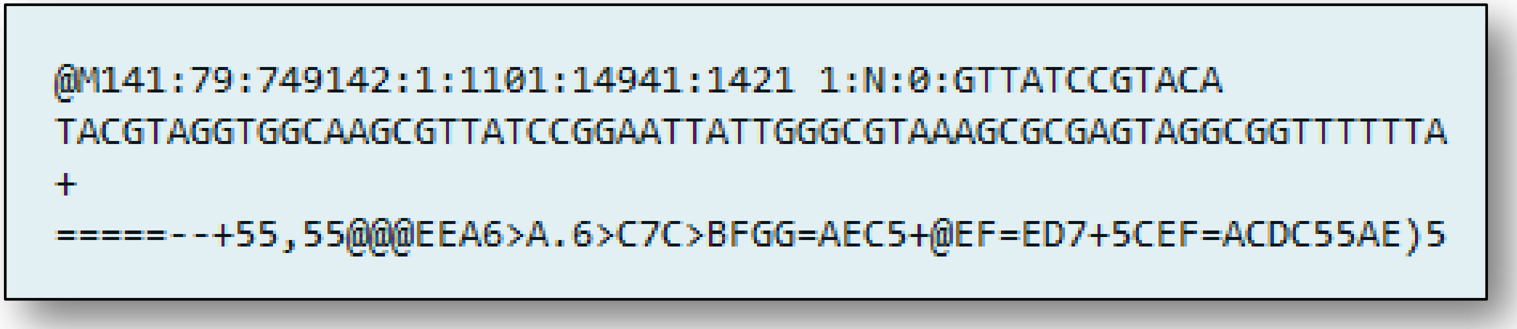
Robert Edgar

Independent scientist
robert@drive5.com
www.drive5.com

FASTQ files

- Text file with four lines per read

1. Label
2. Sequence
3. +
4. Quals



```
@M141:79:749142:1:1101:14941:1421 1:N:0:GTTATCCGTACA
TACGTAGGTGGCAAGCGTTATCCGGAATTATTGGGCGTAAAGCGCAGTAGGCGGTTTTTTA
+
=====--+55,55@@@EEA6>A.6>C7C>BFGG=AEC5+@EF=ED7+5CEF=ACDC55AE)5
```

- Format not fully standardized
  - Different conventions for representing Q scores as letters
  - Software may have different max & min Q scores
  - Typical is Q2 to Q40

Quality (Phred) scores

- Integer Q2 .. Q40
- Represents *P_error*, probability base is wrong

Q40:	*P_error* = 0.0001	**99.99% good**
Q30:	*P_error* = 0.001	99.9% good
Q20:	*P_error* = 0.01	99% good
Q10:	*P_error* = 0.1	10% wrong
Q3:	*P_error* = 0.5	50% wrong
Q2:	*P_error* = 0.66	66% wrong!!

Quality filtering

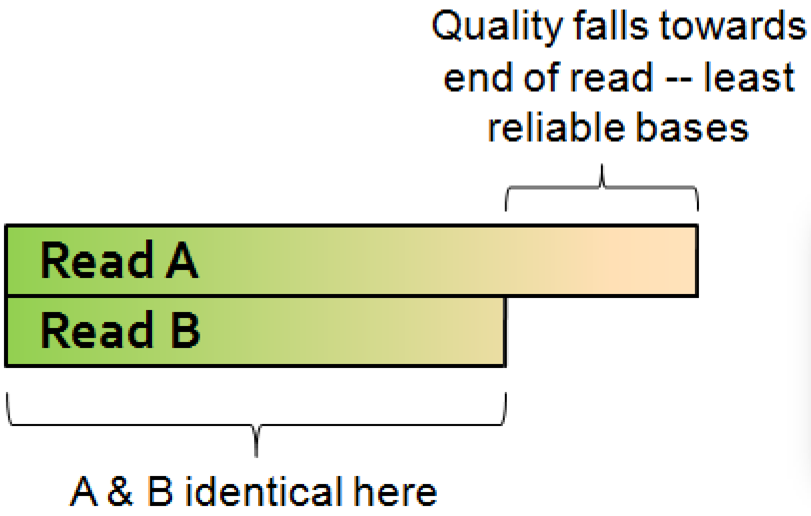
- Discard poor-quality data
- Poor quality = high probability of error(s)
  - low Q scores
- Genomics can mask out low-Q positions
  - e.g. for SNP-calling

Quality filtering

- Amplicon sequencing different scenario
  - Need pair-wise comparisons for most analysis
    - pairs of reads, or reads & database
    - to calculate identity or determine if sequences identical
  - Masked / ambiguous positions (Ns) problematic
  - Variable length (e.g. truncated at low Q) also problematic
- OTU clustering
  - "Harmful" reads >3% errors create spurious OTUs
  - High diversity in harmful reads
  - <u>Many</u> spurious OTUs even if harmful reads small fraction

Truncating at low Q is bad idea

- Read quality often falls towards end of read
- Popular (but bad!) to truncate when Q low



Quality falls towards end of read -- least reliable bases

Read A

Read B

A & B identical here

Do *A* and *B* have identical sequences?

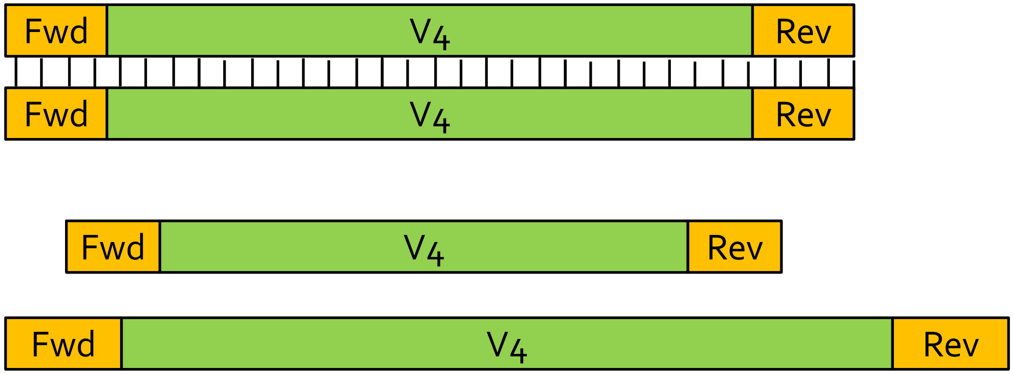
If **Yes**, dubious tail gets high abundance
If **No**, good prefix gets low abundance

Length trimming

- Similar/identical reads should be globally alignable with few/no terminal gaps
- Comparisons unambiguous
  - Cannot have A identical (or >97% similar) to prefix of B
- Unpaired reads: truncate to fixed length
  - Important for 454
  - Often not needed for Illumina
  - Sometimes trim low-quality tails

Global trimming

- Full-length amplicons with varying length ok
  - e.g. overlapping paired reads
  - trim to primers ok
  - no terminal gaps when same / closely related



Quality filtering methods

☑? Minimum Q

- Ok if Q is large, e.g. Q≥20 (*P_error*=1%)
- Ok if don't truncate -- keep or discard

☒ Average Q, maybe over sliding window

- Conceptual mistake -- averaging logarithms!?
- Errors dominated by small Qs

☒ QIIME filter

- Truncate (👎) read if >3 consecutive bases with Q≤3
- Q=3 means *P_error* = 50%
- Allows reads with **many** errors!

Quality filtering methods

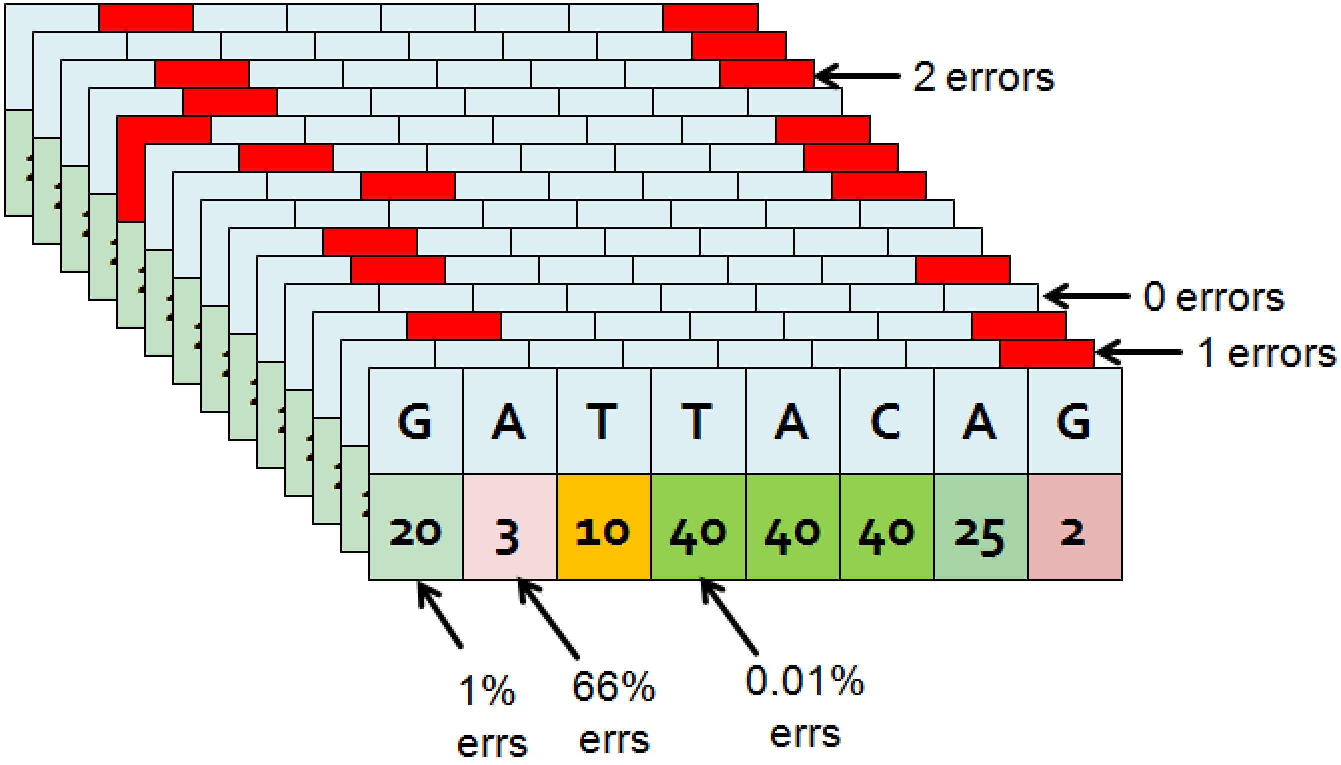
☒ PANDAseq method

- *t* = geometric mean of *P_correct* along read ≥ 0.6
- *P_error* = 0.4
- Much too high, allows reads with many errors
- Better with higher *t*, but not as good as expected errors

☑ Expected error filtering

G	A	T	T	A	C	A	G
20	3	10	40	40	40	25	2

Expected errors



G	A	T	T	A	C	A	G
20	3	10	40	40	40	25	2

2 errors

0 errors

1 errors

1% errs

66% errs

0.01% errs

Expected errors

- Expected errors (*E*) in a read
- *E* = mean over large set with random errors according to Q scores
  - real-valued (because it's an average)
  - always > 0
  - can be < 1

Expected errors

- Surprisingly easy to calculate *E*

  Sum the error probabilities

  E = sum *P_error*

- Most probable number of errors *E**

  *E** = largest integer ≤ *E*

  = *floor*(*E*)

- Proofs in Edgar & Flyvbjerg (2015).

Expected error filter

- Discard reads with *E*>1
  - Keep reads with *E**=0
  - Most probable number of errors = zero
- Typical performance on MiSeq 2x250 V4

keeps 75%+ of the reads

2/3 of filtered reads are correct (zero errors)

1/3 have one or more bad bases

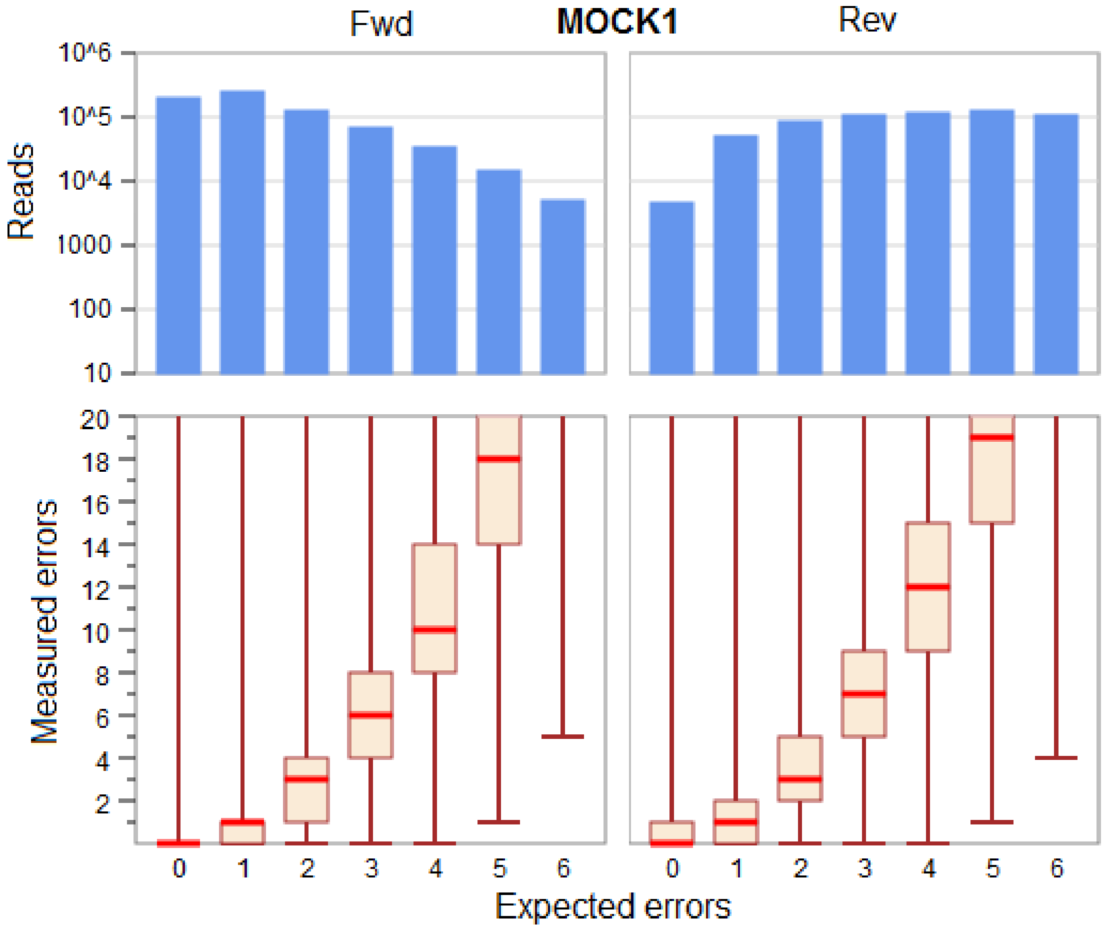
Expected error filter

- Works well if Q scores are accurate
- Illumina Q scores are pretty good
- 454 not so good
  - filtering not so effective
  - expected error filter still best method
- Max *E*=1 suggested default
  - Not a requirement! (note for comparative validation)
  - Larger *E* for less stringent filtering (more spurious OTUs)
  - Smaller *E* for very stringent filtering

Expected error filtering

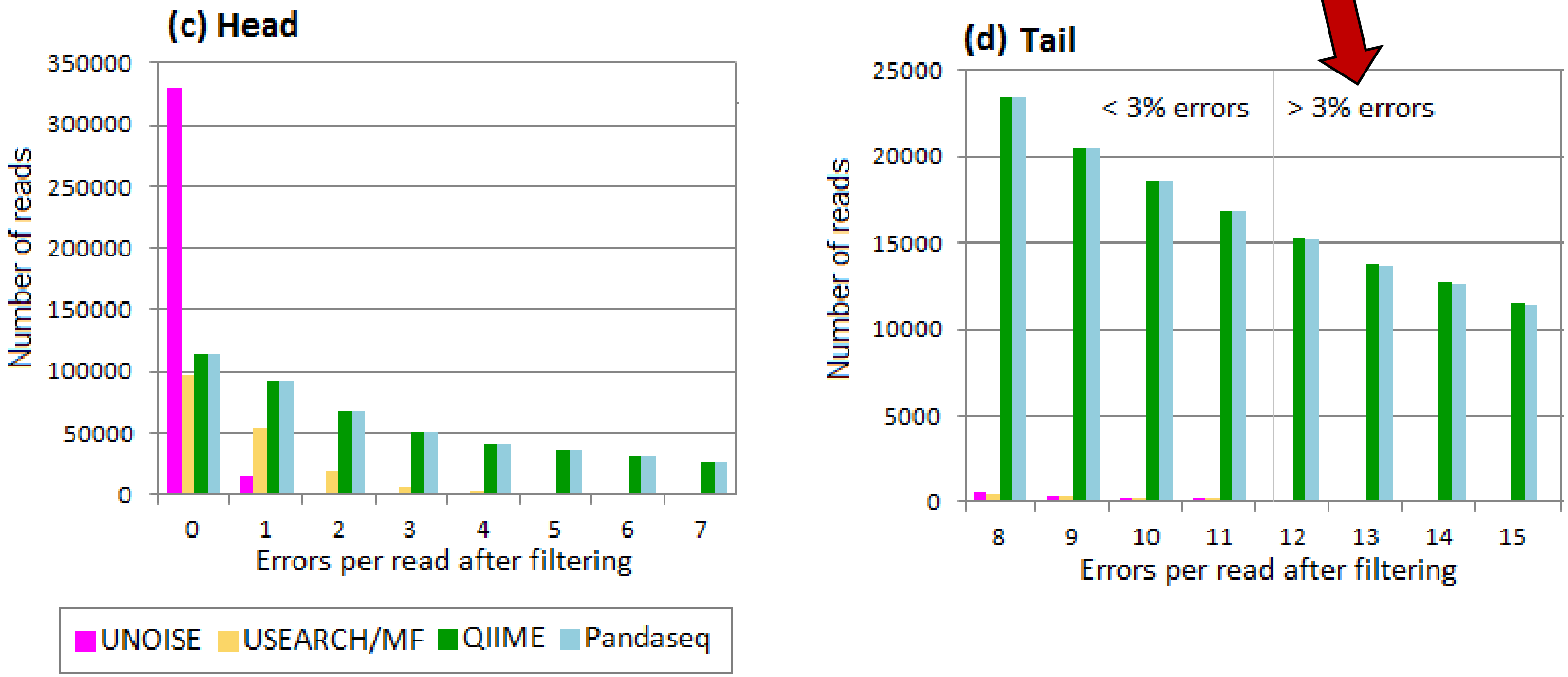
- Critics: allege too stringent
  - high cost in sensitivity, diversity
- Reads are not lost!
  - Most filtered reads map to OTUs after clustering
  - Filtering is critically important to suppress spurious OTUs
- High sensitivity to rare species not possible
  - Contaminants, cross-talk…
  - Limit of resolution abundance > ~0.5% of reads

Expected vs. measured errors

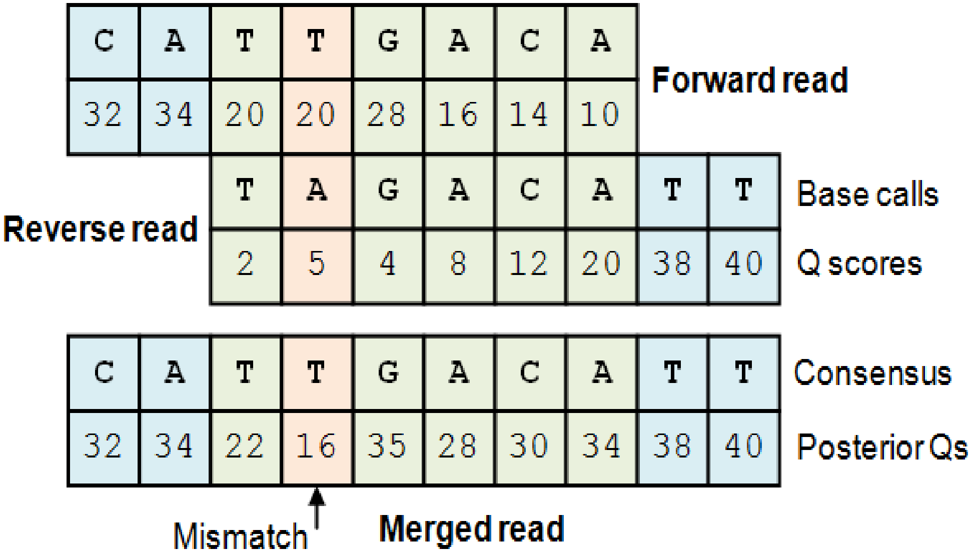


Quality filter performance

QIIME and PANDAseq filters leave tens of thousands of reads with >3% errors, thousands of spurious OTUs

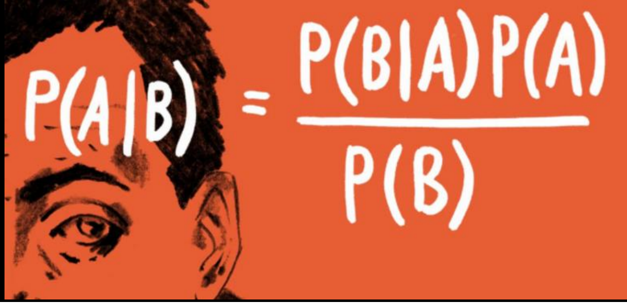


Paired read merging / assembly

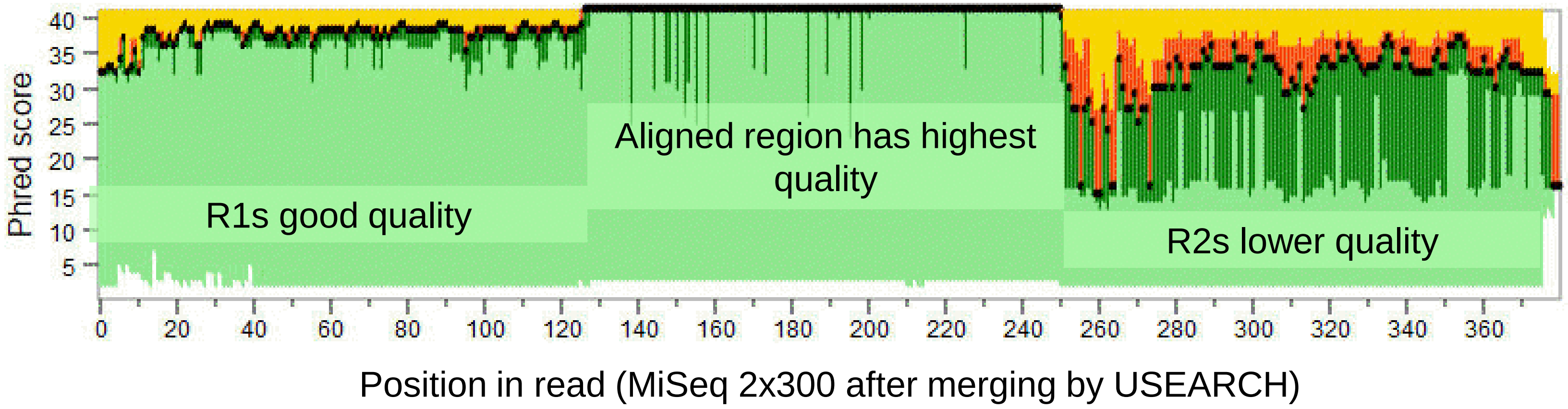


Forward read		C	A	T	T	G	A	C	A		
		32	34	20	20	28	16	14	10		
Reverse read	Base calls			T	A	G	A	C	A	T	T
	Q scores			2	5	4	8	12	20	38	40
Merged read	Consensus	C	A	T	T	G	A	C	A	T	T
	Posterior Qs	32	34	22	16	35	28	30	34	38	40

Paired read merging



- Two observations of each base in overlap
- Should increase/decrease Q if match/mismatch
- Use Bayes' Theorem to get posterior *P_error*
  - Correct equations in Edgar & Flyvbjerg (2015)
  - Previous papers got this wrong!

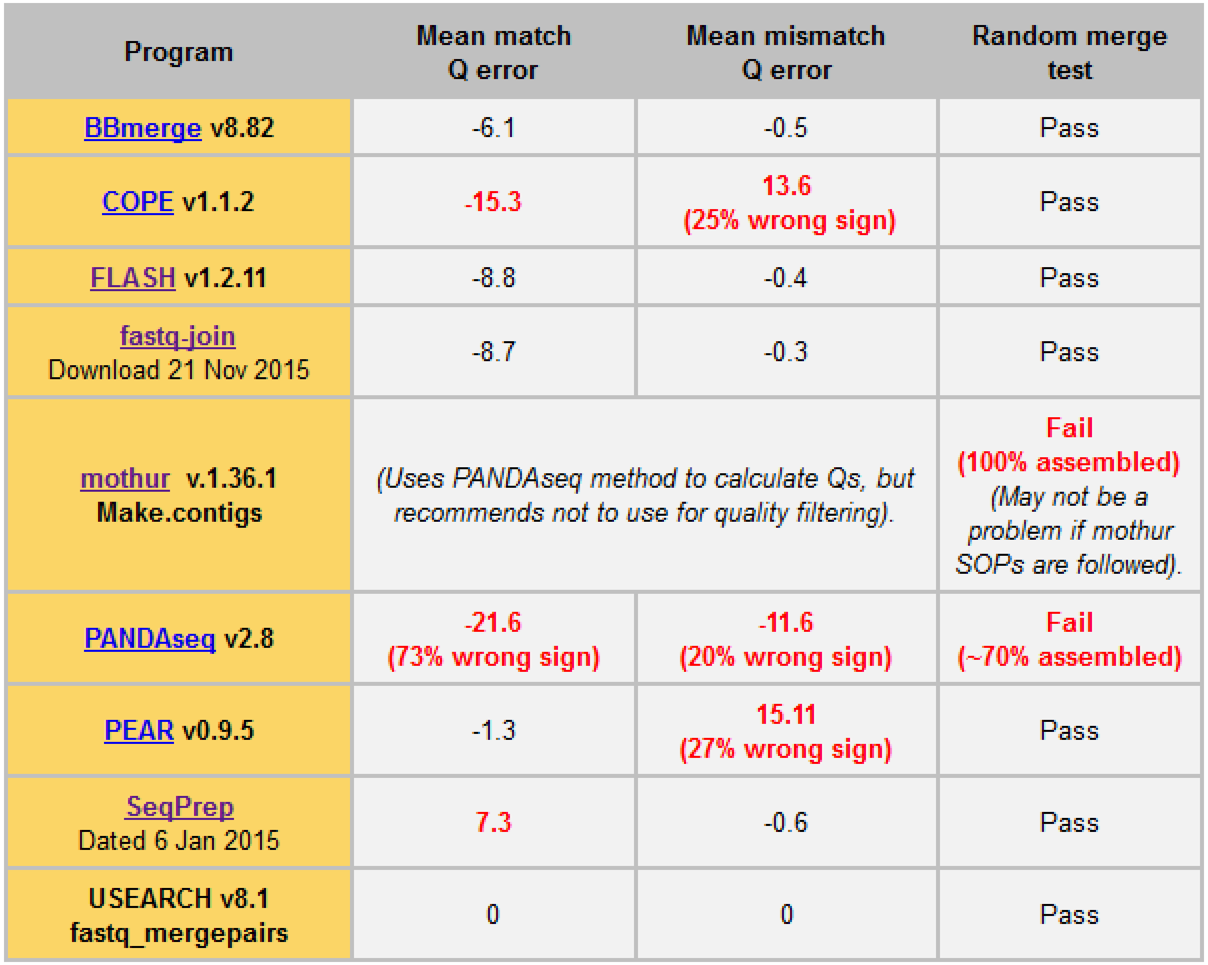


R1s good quality

Aligned region has highest quality

R2s lower quality

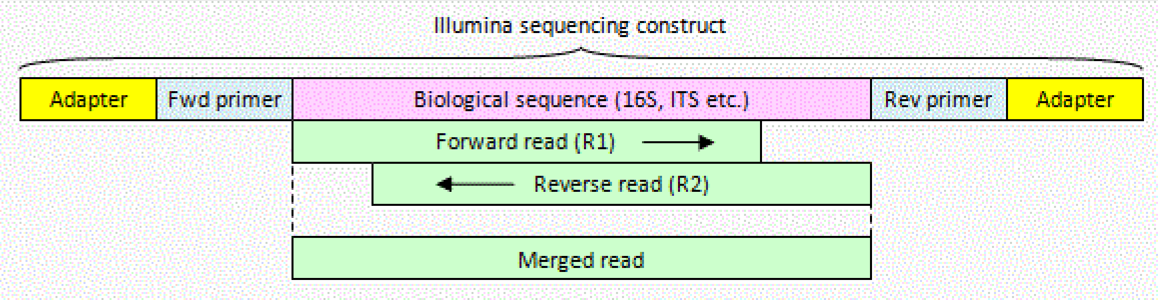
Paired read assemblers



Program	Mean match Q error	Mean mismatch Q error	Random merge test
BBmerge v8.82	-6.1	-0.5	Pass
COPE v1.1.2	-15.3	13.6 (25% wrong sign)	Pass
FLASH v1.2.11	-8.8	-0.4	Pass
fastq-join Download 21 Nov 2015	-8.7	-0.3	Pass
mothur v.1.36.1 Make.contigs	*(Uses PANDAseq method to calculate Qs, but recommends not to use for quality filtering).*		Fail (100% assembled) *(May not be a problem if mothur SOPs are followed).*
PANDAseq v2.8	-21.6 (73% wrong sign)	-11.6 (20% wrong sign)	Fail (~70% assembled)
PEAR v0.9.5	-1.3	15.11 (27% wrong sign)	Pass
SeqPrep Dated 6 Jan 2015	7.3	-0.6	Pass
USEARCH v8.1 fastq_mergepairs	0	0	Pass

Do we need full overlap?

- V4 is ~250nt
- 2x250 PE reads give full overlap
  - Better error correction?
- Accurate OTUs with UPARSE on **R2s only**!
- Longer amplicons ok, e.g. V3-V4 (400nt)
  - better resolution



Dereplication

- Find the unique sequences in the reads
    - and their abundances
- Abundance is a very useful signal
    - Most abundant sequences almost certainly correct
        - unless low-Q truncated
    - Errors increasingly common at lower abundances
- Pool reads from all samples
    - Strongest abundance signal

Singletons

- Abundance = 1
- Random errors usually singletons
  - Not usually reproduced by chance
- Systematic errors may have ab. > 1
  - Polymerase errors & chimeras (amplified by PCR)
  - Sequencing error usually pretty random

Discard singletons

- After filtering, many reads with >3% errors
  - Sequencer error
  - Polymerase copying errors
  - Chimeras
  - Most of these are singletons
- Discard singletons before clustering
  - Necessary to minimize spurious OTUs
  - Most singletons map to OTUs after clustering, not lost!

Discard singletons

- Critics: allege high cost in sensitivity, diversity
- Effect on sensitivity minimal / meaningless
  - By definition, found once in one sample!
  - Ecologically irrelevant (or not possible to interpret)
- Sensitivity is < 100% with singletons
  - Sampling effects, e.g. rare species missed
  - Primer mismatches ("universal" = ~80% - 90%)
  - Some / many rare species missing regardless
- Diversity metrics like Chao1 nonsense for 16S

Delete primer-binding sequences

- PCR tends to substitute mismatches
- Not needed with many Illumina protocols
  - 16S / ITS primer-binding sequence not in read