Topic 3: sequence file formats and quality checking and trimming

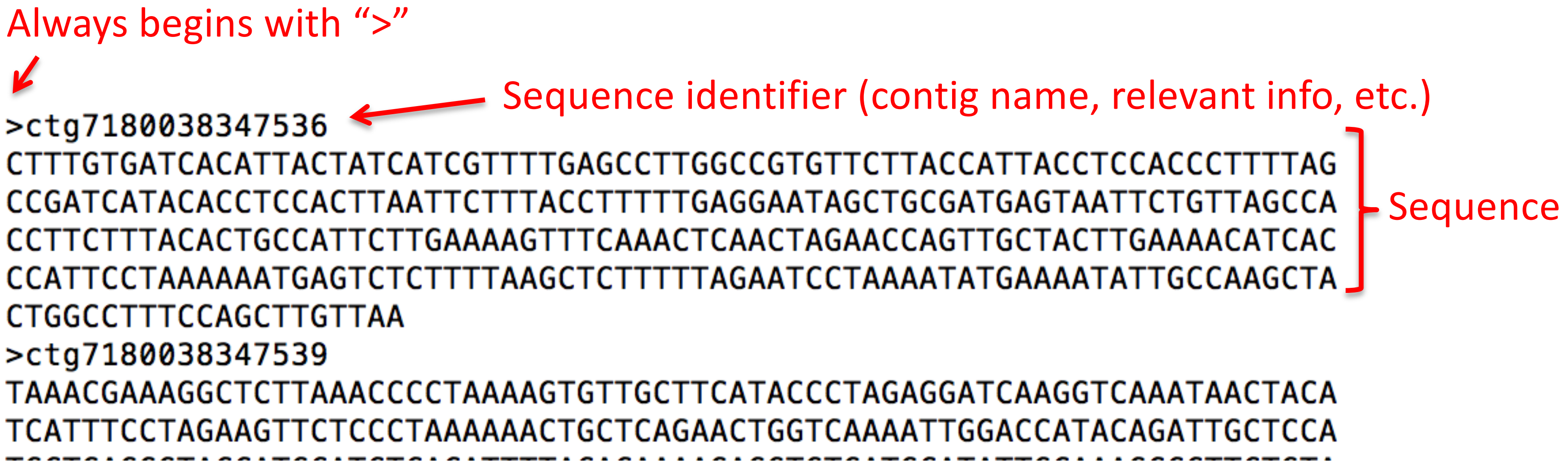
Lecture outcomes

- Understand sequence file formats
- Identify the main steps for preparing NGS data for alignment/assembly

NGS file formats: Fasta

- Sequences with a header (.fasta, .fa, .fas)
- Now mainly used for storing reference sequences (no qual scores) as either nucleotides or peptides
- Can have quality scores are stored in separate files (.qual)

- 2 parts for each sequence:



Always begins with “>”

Sequence identifier (contig name, relevant info, etc.)

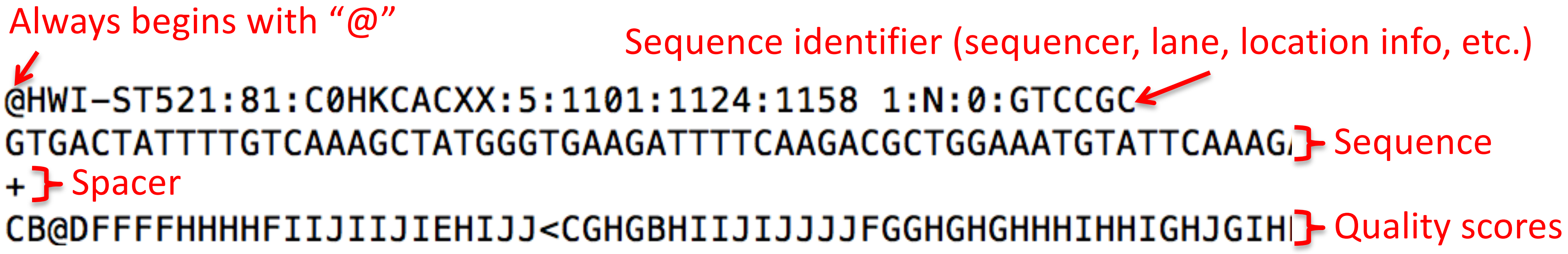
Sequence

```
>ctg7180038347536
CTTTGTGATCACATTACTATCATCGTTTTGAGCCTTGGCCGTGTTCTTACCATTACCTCCACCCTTTTAG
CCGATCATACACCTCCACTTAATTCTTTACCTTTTTGAGGAATAGCTGCGATGAGTAATTCTGTTAGCCA
CCTTCTTTACACTGCCATTCTTGAAAAGTTTCAAACTCAACTAGAACCAGTTGCTACTTGAAAACATCAC
CCATTCCTAAAAAATGAGTCTCTTTTAAGCTCTTTTTAGAATCCTAAAATATGAAAATATTGCCAAGCTA
CTGGCCTTTCCAGCTTGTTAA
>ctg7180038347539
TAAACGAAAGGCTCTTAAACCCCTAAAAGTGTTGCTTCATACCCTAGAGGATCAAGGTCAAATAACTACA
TCATTTCCTAGAAGTTCTCCCTAAAAAACTGCTCAGAACTGGTCAAAATTGGACCATACAGATTGCTCCA
```

NGS file formats: Fastq

FASTQ:

- Sequence and quality scores are stored in the same file (usually .fq or .fastq)
- Most common format for short read data returned from the sequencer
- 4 lines/sequence read:



```
@HWI-ST521:81:C0HKCACXX:5:1101:1124:1158 1:N:0:GTCCGC
GTGACTATTTTGTCAAAGCTATGGGTGAAGATTTTCAAGACGCTGGAAATGTATTCAAAG
+
CB@DFFFFHHHHFIIJIIJIEHIJJ<CGHGBHIIJIJJJJFGGHGHGHHHIHHIGHJGIH
```

Always begins with “@”

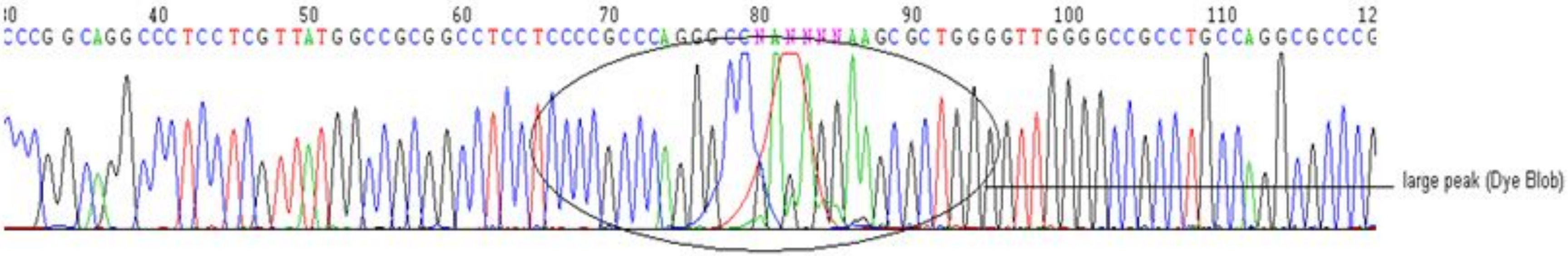
Sequence identifier (sequencer, lane, location info, etc.)

Sequence

Spacer

Quality scores

NGS file formats: Quality scores

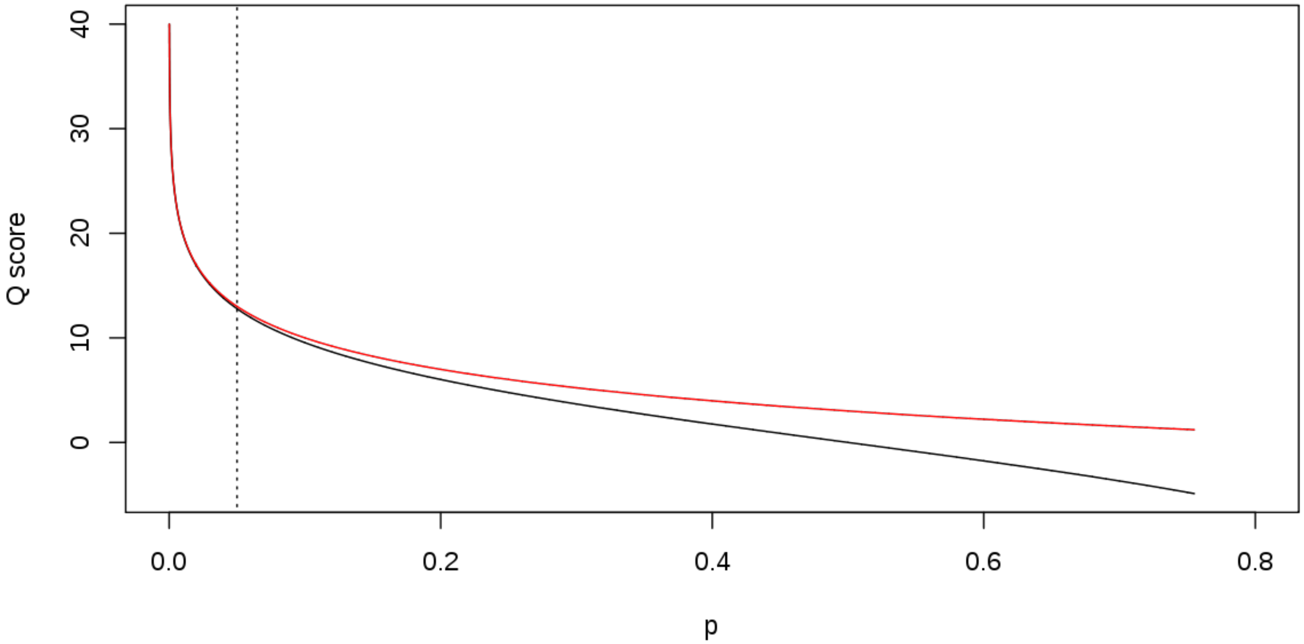


NGS file formats: Quality scores

Historically, two formats (now all are Sanger)

- $Q_{sanger} = -10 * \log_{10}(p)$
- $Q_{solexa} = -10 * \log_{10}(p / (1 - p))$

where p is the probability that a base call is incorrect



High quality scores are good

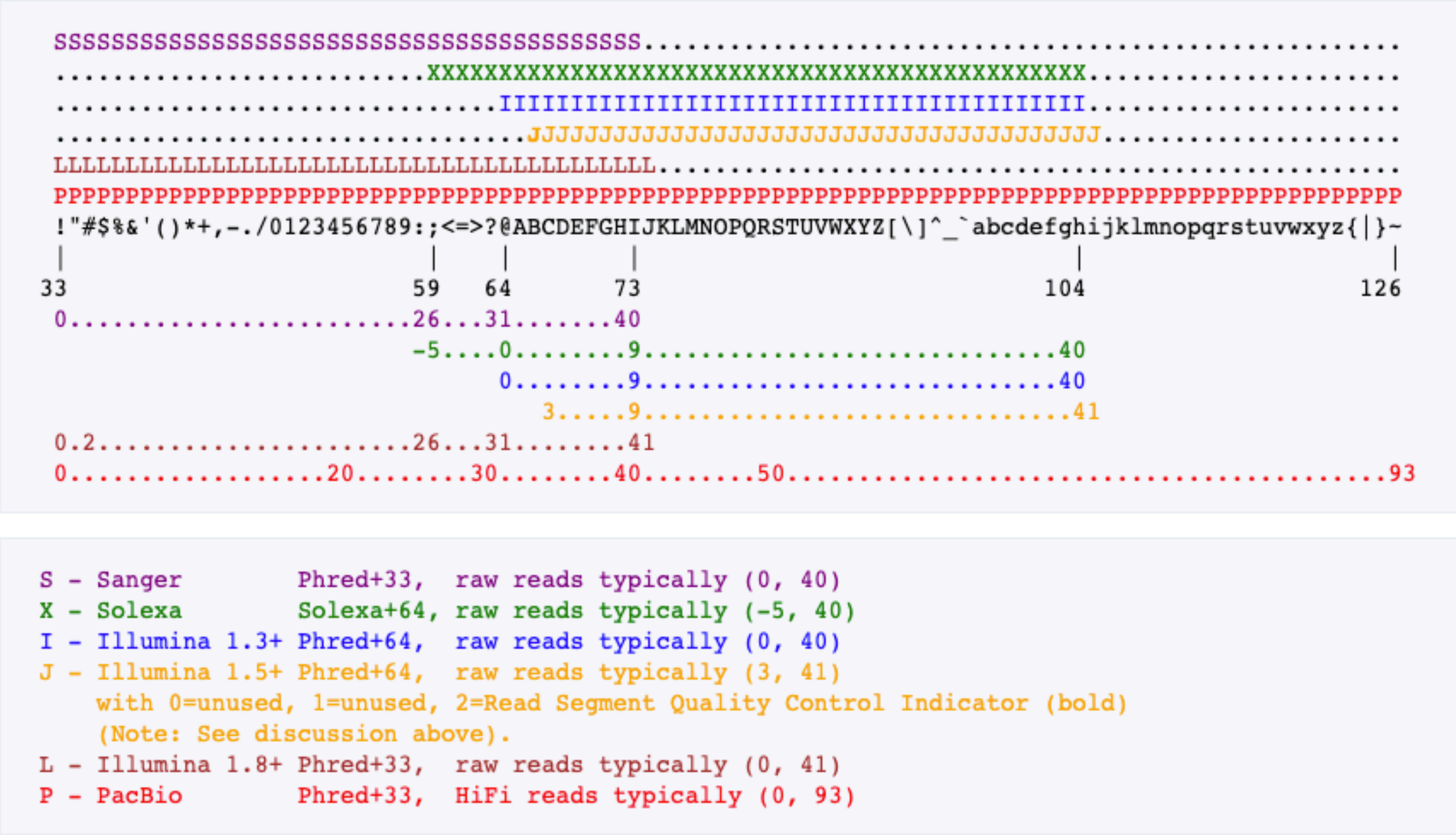
To calculate p from Q:

p = 10^(-Q / 10)

Q30 = 0.1% p[incorrect]
Q20 = 1% p[incorrect]
Q10 = 10% p[incorrect]

http://en.wikipedia.org/wiki/FASTQ_format

NGS file formats: Quality scores



```
 SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS.....................................................
 ..........................XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX......................
 ...............................IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII......................
 .................................JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ.....................
 LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL....................................................
 PPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPPP
 !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
 |                         |    |        |                              |                     |
33                        59   64       73                            104                   126
 0........................26...31.......40
                          -5....0........9.............................40
                                0........9.............................40
                                   3.....9..............................41
 0.2......................26...31........41
 0..................20........30........40........50..........................................93
```

```
S - Sanger        Phred+33,  raw reads typically (0, 40)
X - Solexa        Solexa+64, raw reads typically (-5, 40)
I - Illumina 1.3+ Phred+64,  raw reads typically (0, 40)
J - Illumina 1.5+ Phred+64,  raw reads typically (3, 41)
    with 0=unused, 1=unused, 2=Read Segment Quality Control Indicator (bold)
    (Note: See discussion above).
L - Illumina 1.8+ Phred+33,  raw reads typically (0, 41)
P - PacBio        Phred+33,  HiFi reads typically (0, 93)
```

Fortunately, we seem to have settled on a standard in the community…for now!

http://en.wikipedia.org/wiki/FASTQ_format

Code break

There are more unix examples at the end of Github Topic 3 page (or README.txt file in the ~/Topic_3 folder)

1) How many sequences do you have in the file ~/Topic_3/data/Pine_reference_rnaseq_reduced.fa?

Hint: wc –l <file name> provides the number of lines in a file

2) How many sequences do you have in the fastq file ~/Topic_3/data/GBS12_brds_Pi_197A2_100k_R1.fastq?

Hint: for grep ^ indicates the start of the line and $ indicates the end of the line (e.g. grep ^H*?$ <filename> would find all the lines starting with H and ending in ?)

3) How many sequences contain a base with a Phred score of 2 ~/Topic_3/data/GBS12_brds_Pi_197A2_100k_R1.fastq?

Preparing Fastq for analysis

1) Check files for completeness, use md5 checksums if file corruption is suspected

2) Inspect quality statistics

3) Possible steps to clean files (choice of steps depends on the application)

- De-multiplex
- Trim adapters
- Filter/trim low quality base calls
- Remove duplicate sequences
- Remove contaminant sequences
- Remove sequences that are mainly adapter

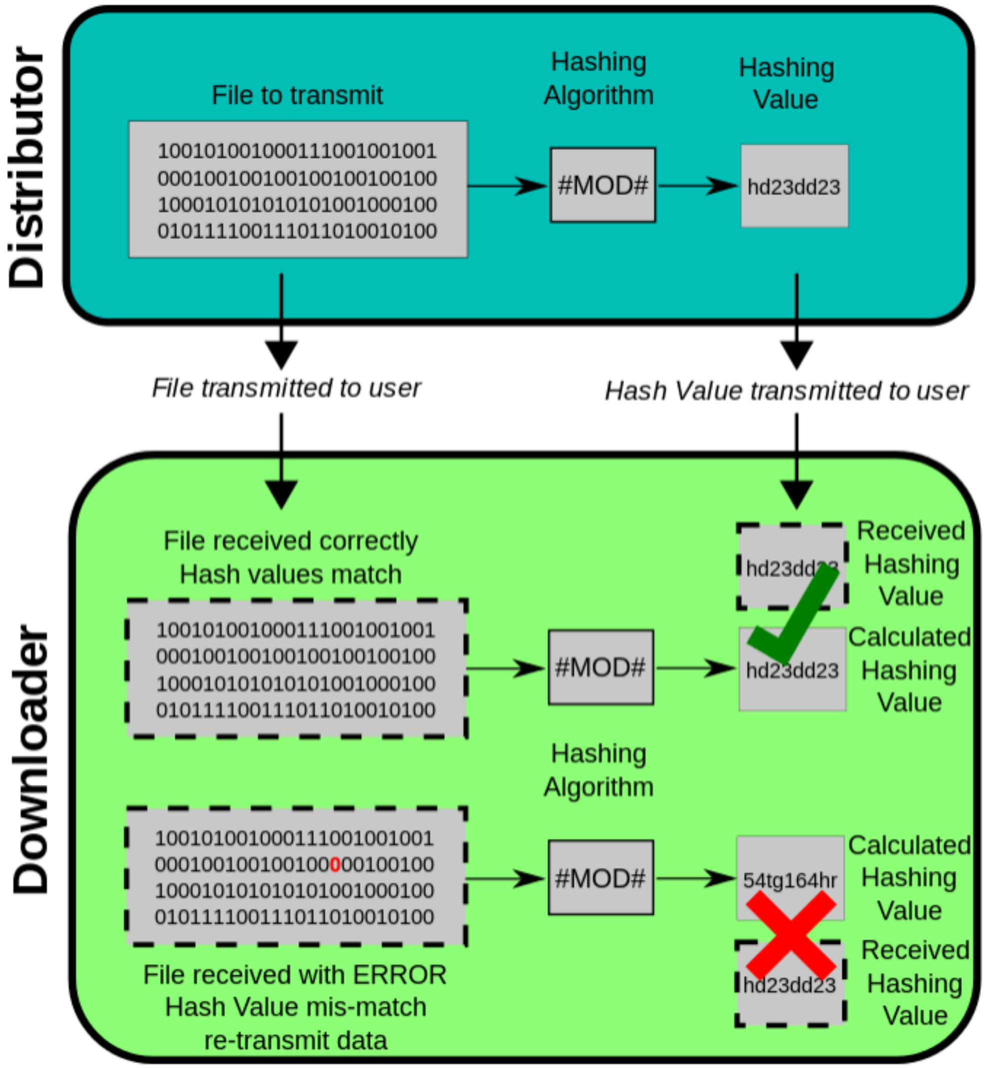
Usually done by sequencing center

Genotyping and RNAseq

Reference assembly

Many programs to implement these steps!

Preparing Fastq: md5 checksum



Distributor

File to transmit

100101001000111001001001
000100100100100100100100
100010101010101001000100
010111100111011010010100

Hashing Algorithm

#MOD#

Hashing Value

hd23dd23

File transmitted to user

Hash Value transmitted to user

Downloader

File received correctly
Hash values match

100101001000111001001001
000100100100100100100100
100010101010101001000100
010111100111011010010100

#MOD#

hd23dd23

Received Hashing Value

hd23dd23

Calculated Hashing Value

Hashing Algorithm

100101001000111001001001
000100100100100000100100
100010101010101001000100
010111100111011010010100

#MOD#

54tg164hr

Calculated Hashing Value

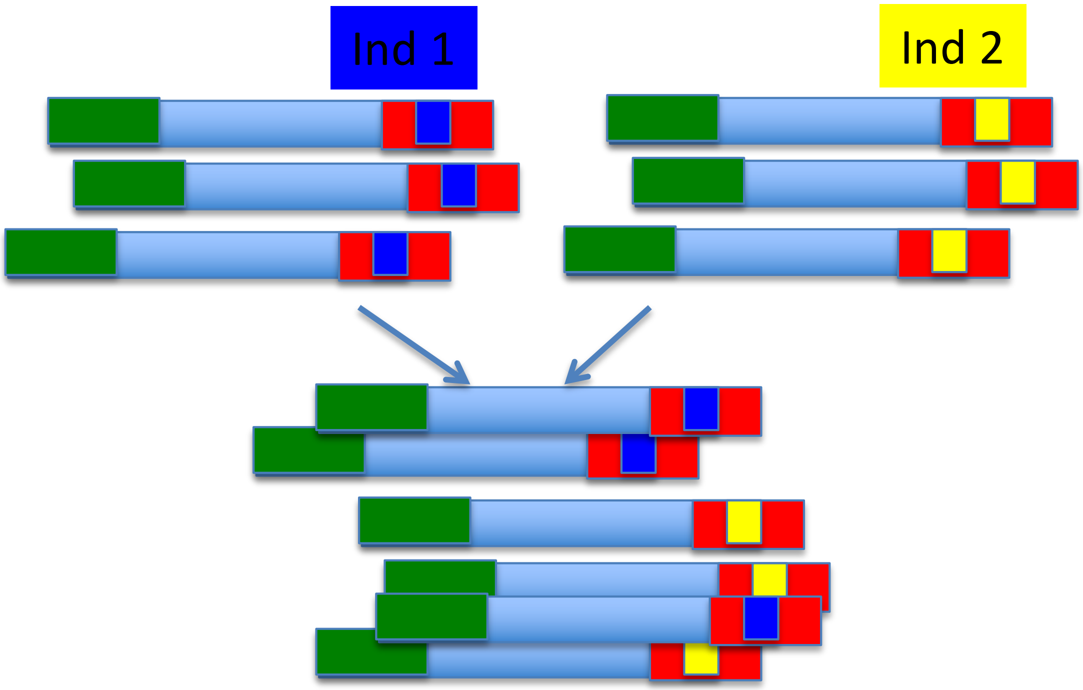
hd23dd23

Received Hashing Value

File received with ERROR
Hash Value mis-match
re-transmit data

Preparing Fastq: De-multiplexing

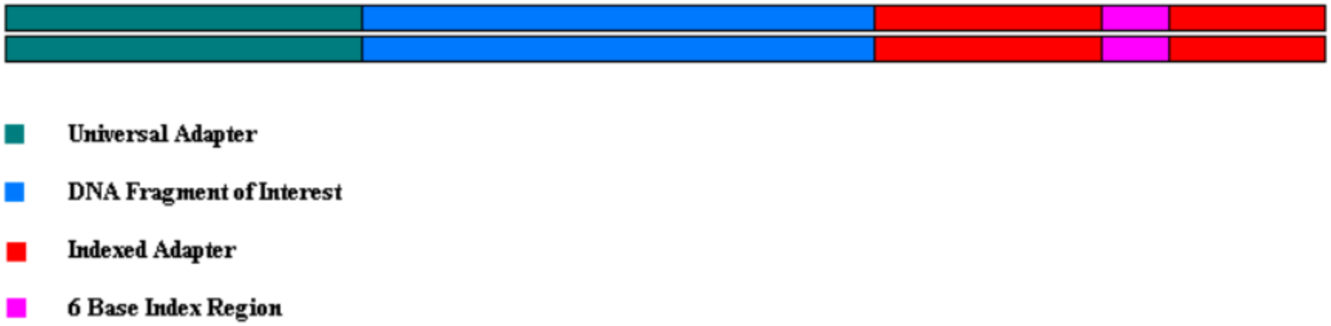
Multiplexing is when several libraries are barcoded and sequenced on the same lane



- Most sequencing centers will de-multiplex the data
- Casava can be used for de-multiplexing and trimming barcodes from standard Illumina library preps

Preparing Fastq: Trimming

- Adapters are short sequences that are added to the beginning and end of DNA molecules to prepare them for sequencing



- Can compromise how well the reads align to a reference if not removed
- Detect during the quality control phase
- Removed by a range of tools (most sequencing centers will already have removed the adapters)

Preparing Fastq: Quality metrics

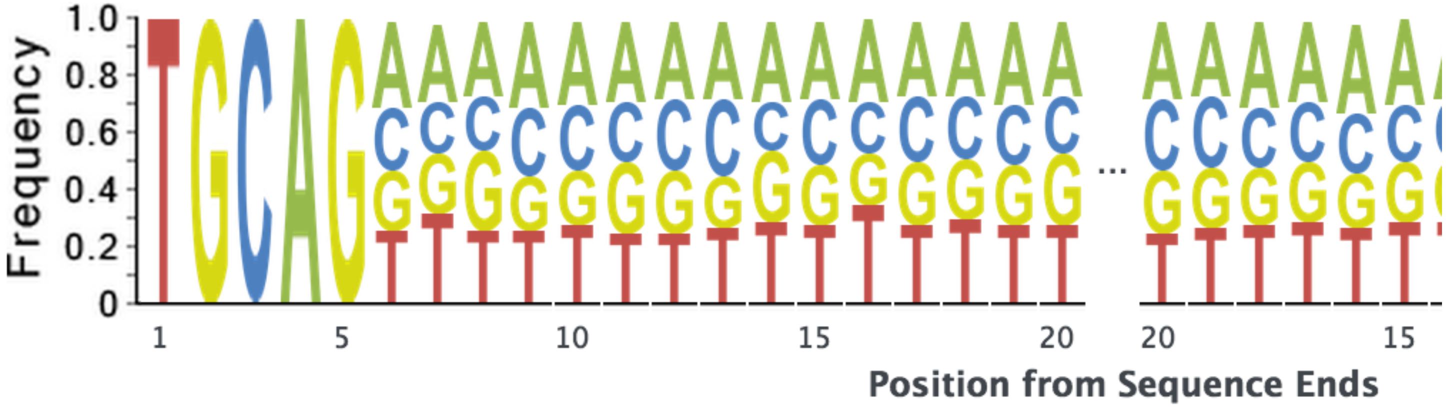
Many possible statistics to query:

- Number and length of sequences
- Base qualities
- Poly A/T tails
- Presence of tag sequences (stuff you added during preparation)
- Sequence complexity (e.g. ATATATATATATA…)

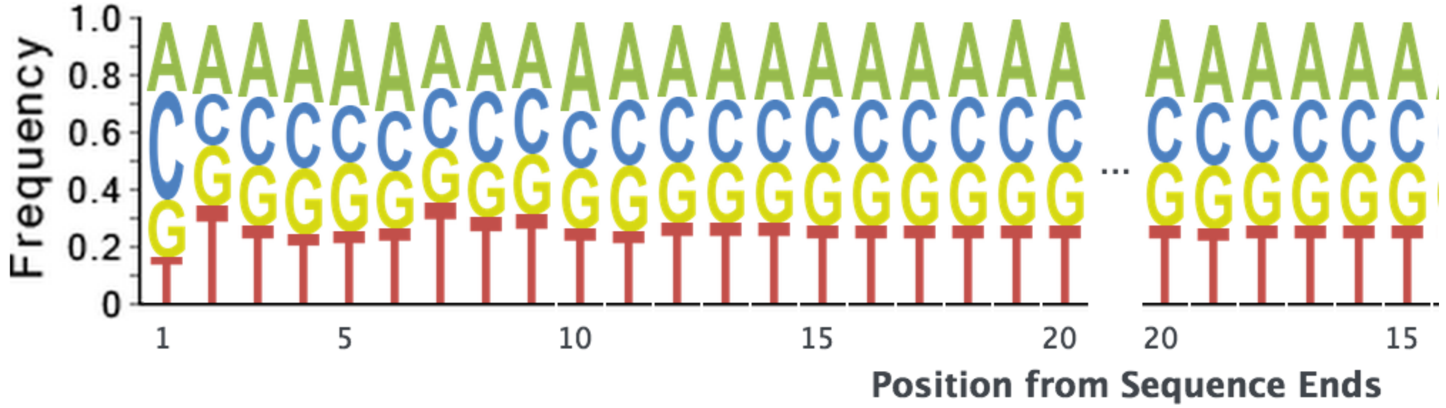
Recommended tools: prinseq, fastqc

Preparing Fastq: Quality metrics

Distribution of base frequencies in GBS reads with enzyme cut site:

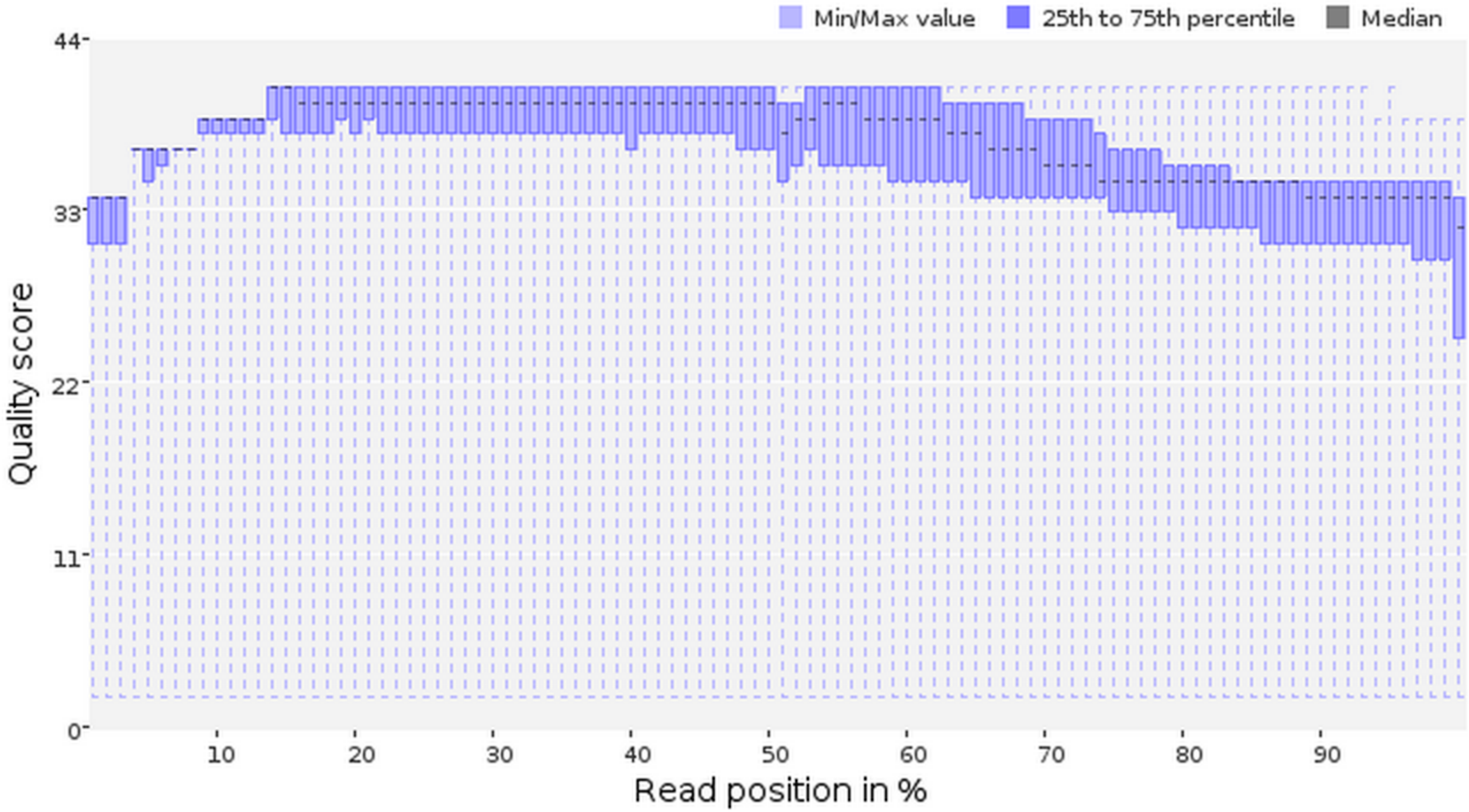


Distribution in RNAseq data, no adapters/tags:



Preparing Fastq: Quality metrics

A normal quality score distribution for Illumina reads:



Preparing Fastq: Quality trimming

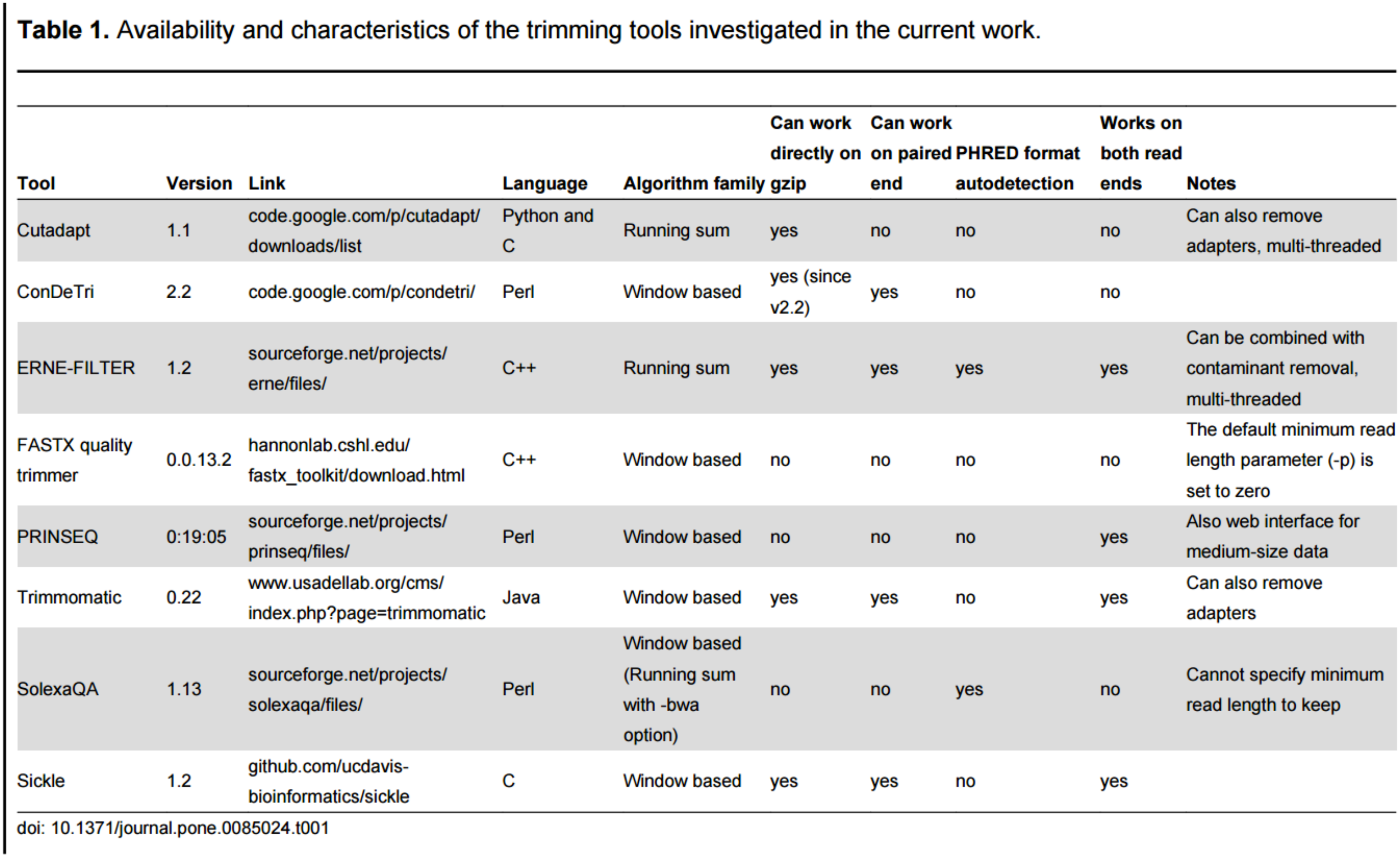


Table 1. Availability and characteristics of the trimming tools investigated in the current work.

Tool	Version	Link	Language	Algorithm family	Can work directly on gzip	Can work on paired end	PHRED format autodetection	Works on both read ends	Notes
Cutadapt	1.1	code.google.com/p/cutadapt/downloads/list	Python and C	Running sum	yes	no	no	no	Can also remove adapters, multi-threaded
ConDeTri	2.2	code.google.com/p/condetri/	Perl	Window based	yes (since v2.2)	yes	no	no	
ERNE-FILTER	1.2	sourceforge.net/projects/erne/files/	C++	Running sum	yes	yes	yes	yes	Can be combined with contaminant removal, multi-threaded
FASTX quality trimmer	0.0.13.2	hannonlab.cshl.edu/fastx_toolkit/download.html	C++	Window based	no	no	no	no	The default minimum read length parameter (-p) is set to zero
PRINSEQ	0:19:05	sourceforge.net/projects/prinseq/files/	Perl	Window based	no	no	no	yes	Also web interface for medium-size data
Trimmomatic	0.22	www.usadellab.org/cms/index.php?page=trimmomatic	Java	Window based	yes	yes	no	yes	Can also remove adapters
SolexaQA	1.13	sourceforge.net/projects/solexaqa/files/	Perl	Window based (Running sum with -bwa option)	no	no	yes	no	Cannot specify minimum read length to keep
Sickle	1.2	github.com/ucdavis-bioinformatics/sickle	C	Window based	yes	yes	no	yes	

doi: 10.1371/journal.pone.0085024.t001

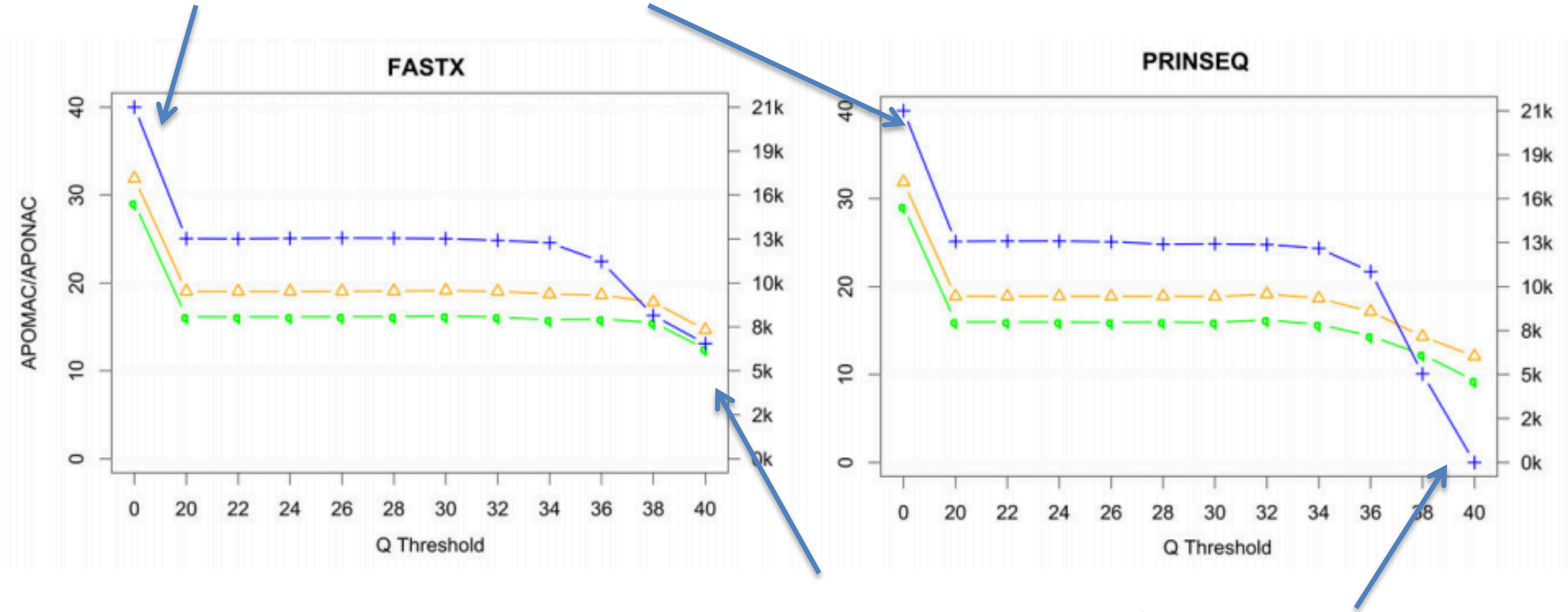
Preparing Fastq: Quality trimming

Choice of quality score to filter to depends upon the application:

- Too low a quality score cutoff:
  1) increase run times and RAM usage
  2) bad results (e.g. false SNP calls)
- Too high a quality score cutoff:
  1) faster run times
  2) lose useful data (e.g. more fragmented assemblies, missing SNPs)
- Usually Q20, but sometimes lower or higher

Preparing Fastq: Quality trimming

Blue line = SNP number
no trimming – many false SNPs

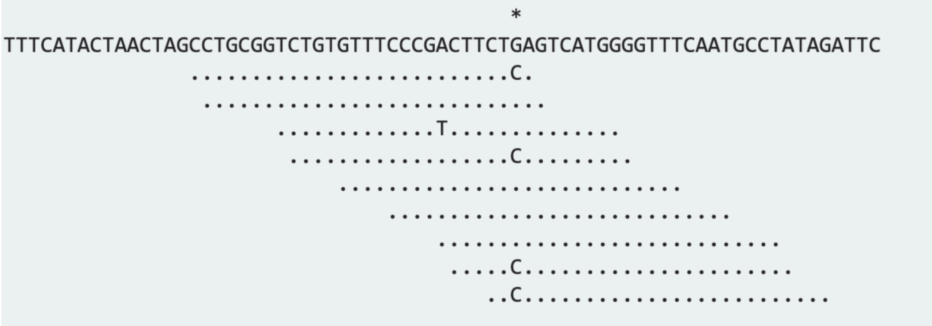


severe trimming - many fewer SNPs

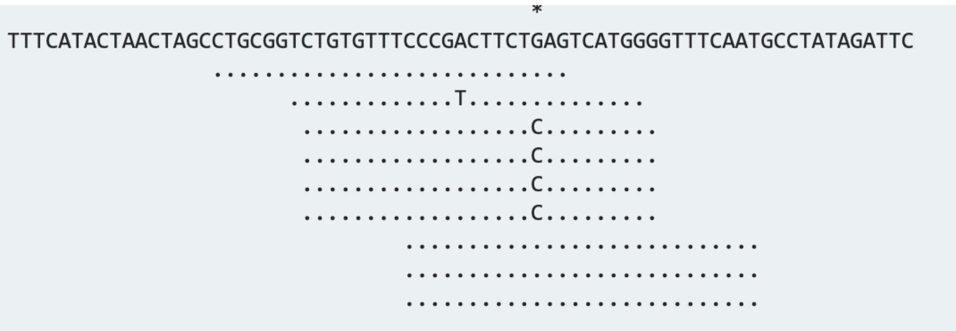
Preparing Fastq: Duplicate identification

- PCR is a common feature of many library preps
- Can introduce errors and biases that can impact downstream analysis
- High % duplicates usually is a sign of wasted sequencing effort
- However, high duplicates rates are expected in some cases

(library and depth dependent) and should not be removed (e.g., GBS, RNAseq)



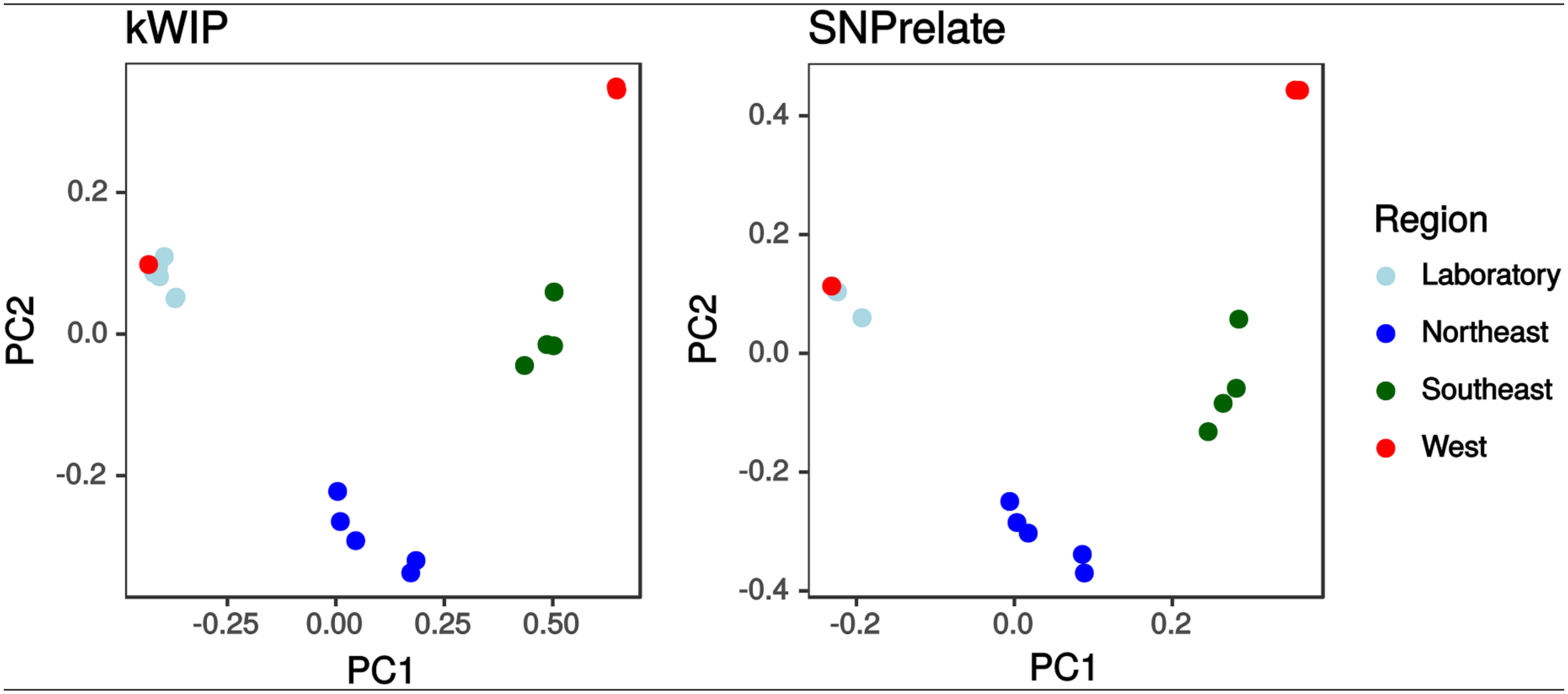
```
                                         *
TTTCATACTAACTAGCCTGCGGTCTGTGTTTCCCGACTTCTGAGTCATGGGGTTTCAATGCCTATAGATTC
               ..........................C.
                ............................
                      .............T..............
                       ..................C.........
                           ............................
                               ............................
                                   ............................
                                    .....C......................
                                       ..C.........................
```



```
                                         *
TTTCATACTAACTAGCCTGCGGTCTGTGTTTCCCGACTTCTGAGTCATGGGGTTTCAATGCCTATAGATTC
                ............................
                      .............T..............
                       ..................C.........
                       ..................C.........
                       ..................C.........
                       ..................C.........
                               ............................
                               ............................
                               ............................
```

Preparing Fastq: Contamination

- Checking for sample contamination using trimmed/filtered reads and alignment free estimators of genetic distance (e.g., kWip)



Chlamydomonas reinhardtii

Preparing Fastq: Assembly

- Remove sequences consisting of adapter dimers (otherwise, they may be included as contigs). (e.g. tagdust)
- Clean out contaminants by blasting to known databases (can also be conducted post-assembly)
- Remove duplicate sequences: for *de novo* assembly, sequences that are exact copies will slow down the assembly without adding anything (e.g. fastx_collapser)

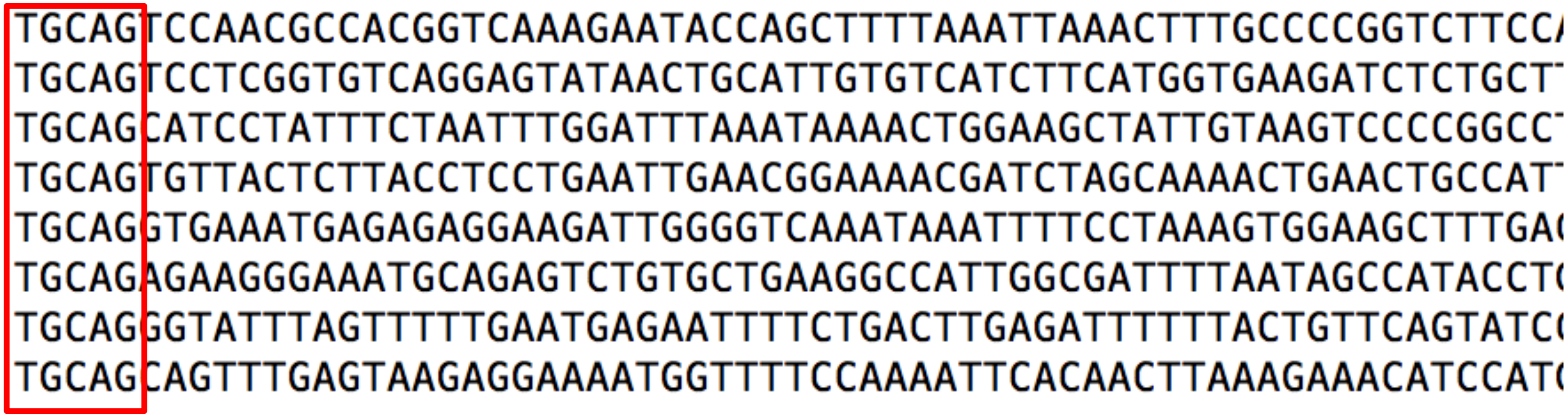
Preparing Fastq: Pairing

- With paired-end reads, if one read direction is removed but the other is not, then the _R1 and _R2 files are mismatched
- Need to run a script to eliminate unpaired reads from each _R1 and _R2 file

Some programs output reads in paired and unpaired files (e.g. prinseq, Trimmomatic). Others do not and custom scripts are required to re-pair data.

Preparing Fastq: GBS-specific filtering

- GBS / RAD use enzymes to cleave the DNA, so all reads will begin with the recognition sequence:



```
TGCAGTCCAACGCCACGGTCAAAGAATACCAGCTTTTAAATTAAACTTTGCCCCGGTCTTCC
TGCAGTCCTCGGTGTCAGGAGTATAACTGCATTGTGTCATCTTCATGGTGAAGATCTCTGCT
TGCAGCATCCTATTTCTAATTTGGATTTAAATAAAACTGGAAGCTATTGTAAGTCCCCGGCC
TGCAGTGTTACTCTTACCTCCTGAATTGAACGGAAAACGATCTAGCAAAACTGAACTGCCAT
TGCAGGTGAAATGAGAGAGGAAGATTGGGGTCAAATAAATTTTCCTAAAGTGGAAGCTTTGA
TGCAGAGAAGGGAAATGCAGAGTCTGTGCTGAAGGCCATTGGCGATTTTAATAGCCATACCT
TGCAGGGTATTTAGTTTTTGAATGAGAATTTTCTGACTTGAGATTTTTTACTGTTCAGTATC
TGCAGCAGTTTGAGTAAGAGGAAAATGGTTTTCCAAAATTCACAACTTAAAGAAACATCCAT
```

- Will need to de-multiplex using Axe, Stacks or custom script
- Clean GBS-specific adapters or other home-brew sequences that sequencing centers didn’t remove

Further reading

- Del Fabbro et al. 2013. An Extensive Evaluation of Read Trimming Effects on Illumina NGS Data Analysis. PLoSOne. 8:e85024.

- http://prinseq.sourceforge.net/Data_preprocessing.pdf

- http://prinseq.sourceforge.net/manual.html#STANDALONE