```
  SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS.....................................................
  ..........................XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX......................
  ..............................IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII......................
  ................................JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ......................
  ..LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL.....................................................
  !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
  |                          |    |        |                              |                     |
 33                         59   64       73                             104                   126
  0.........................26...31.......40
                            -5....0........9.............................40
                                 0........9.............................40
                                    3.....9.............................40
  0.2.......................26...31........41

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, 40)
    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)
```

Q1: Based on the the above, which of characters are consistent with which encoding?

a) `*>@A`

b) `<P[^`

c) `EJ\@`

Q1

MANUAL Scythe can be run minimally with:

```
scythe -a adapter_file.fasta -o trimmed_sequences.fasta sequences.fastq
```

Run scythe the adaptor file the outfile the infile

By default, Illumina's quality scheme (pipeline > 1.3) is used. Sanger or Solexa (pipeline < 1.3) qualities can be specified with `-q`:

```
scythe -a adapter_file.fasta -q solexa -o trimmed_sequences.fasta sequences.fastq
```

Q2: Based on the the above – indicate which words go in which box.

[ ]/scythe –a [ ] –q [ ] –o [ ] [ ]

a) $dir_scy (variable with the path to the software scythe)
b) 1_scy_human001_pe1.fastq (outfile name)
c) adaptors.fa
d) human001_pe1.fastq
e) sanger

Q2

MANUAL Scythe can be run minimally with:

```
scythe -a adapter_file.fasta -o trimmed_sequences.fasta sequences.fastq
```

Run scythe the adaptor file the outfile the infile

By default, Illumina's quality scheme (pipeline > 1.3) is used. Sanger or Solexa (pipeline < 1.3) qualities can be specified with `-q`:

```
scythe -a adapter_file.fasta -q solexa -o trimmed_sequences.fasta sequences.fastq
```

Q2: Based on the the above – indicate which words go in which box.

```
$dir_scy/scythe –a adaptors.fa –q sanger –o 1_scy_human001_pe1.fastq human001_pe1.fastq
```

a) $dir_scy (variable with the path to the software scythe)
b) 1_scy_human001_pe1.fastq (outfile name)
c) adaptors.fa
d) human001_pe1.fastq
e) sanger

Q2

MANUAL Scythe can be run minimally with:

```
scythe -a adapter_file.fasta -o trimmed_sequences.fasta sequences.fastq
```

Run scythe the adaptor file the outfile the infile

By default, Illumina's quality scheme (pipeline > 1.3) is used. Sanger or Solexa (pipeline < 1.3) qualities can be specified with `-q` :

```
scythe -a adapter_file.fasta -q solexa -o trimmed_sequences.fasta sequences.fastq
```

Q2: Based on the the above – indicate which words go in which box.

[ ]/scythe –a [ ] –q [ ] –o [ ] [ ]

a) $dir_scy (variable with the path to the software scythe)
b) 1_scy_human001_pe2.fastq (outfile name)
c) adaptors.fa
d) human001_pe2.fastq
e) sanger

Q2

MANUAL Scythe can be run minimally with:

```
scythe -a adapter_file.fasta -o trimmed_sequences.fasta sequences.fastq
```

Run scythe the adaptor file the outfile the infile

By default, Illumina's quality scheme (pipeline > 1.3) is used. Sanger or Solexa (pipeline < 1.3) qualities can be specified with `-q`:

```
scythe -a adapter_file.fasta -q solexa -o trimmed_sequences.fasta sequences.fastq
```

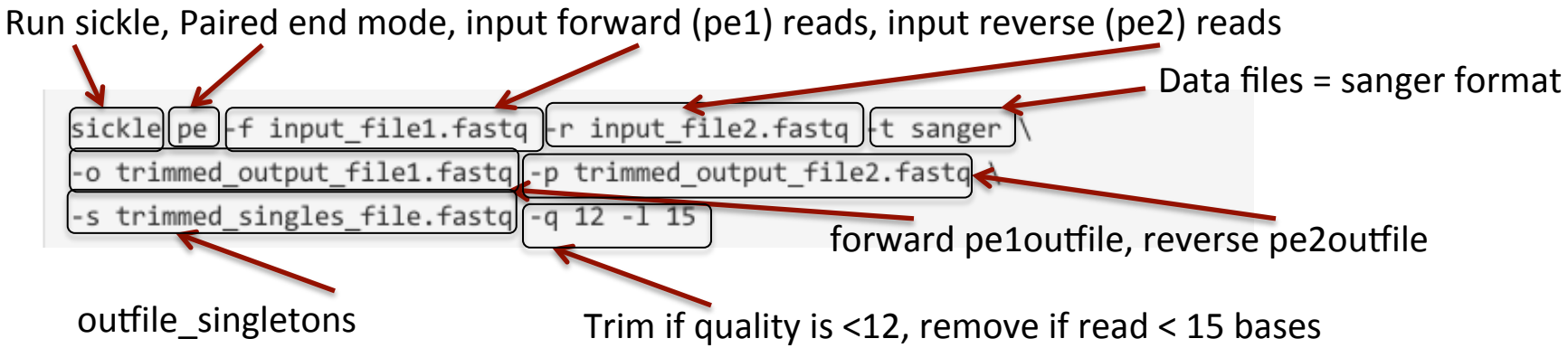
Q2: Based on the the above – indicate which words go in which box.

```
$dir_scy/scythe –a adaptors.fa –q sanger –o 1_scy_human001_pe2.fastq human001_pe2.fastq
```

a) $dir_scy (variable with the path to the software scythe)
b) 1_scy_human001_pe2.fastq (outfile name)
c) adaptors.fa
d) human001_pe2.fastq
e) sanger

Q2

MANUAL



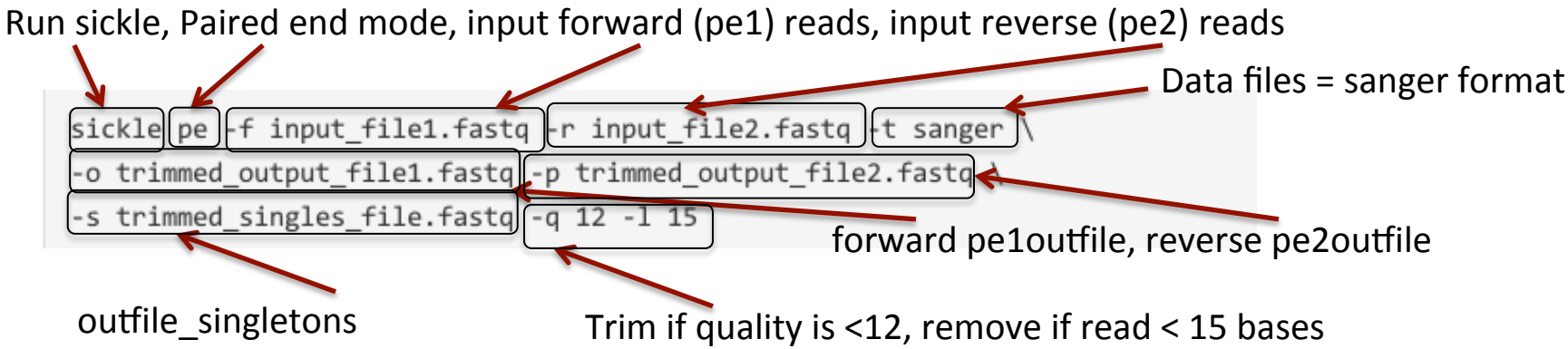
Q3: Based on the the above – indicate which words go in which box. We want to trim to a quality of 20 and length of 20

[ ] sickle [ ] -f [ ] -r [ ] -t [ ] -o [ ] -p [ ] -s [ ] -q [ ] -l [ ]

a) sanger
b) 1_scy_human001_pe1.fastq
c) 1_scy_human001_pe2.fastq
d) $dir_sic
e) 2_sic_human001_pe1.fastq
f) 2_sic_human001_pe2.fastq
g) 2_sic_human001_sing.fastq
h) 20
i) pe
j) 20

Q2

MANUAL



Q3: Based on the the above – indicate which words go in which box. We want to trim to a quality of 20 and length of 20

```
$dir_sic/sickle pe –f 1_scy_human001_pe1.fastq -r 1_scy_human001_pe1.fastq -t sanger
-o 2_sic_human001_pe1.fastq –p 2_sic_human001_pe2.fastq
-s 2_sic_human001_sing.fastq -q 20 –l 20
```

Q2

BWA

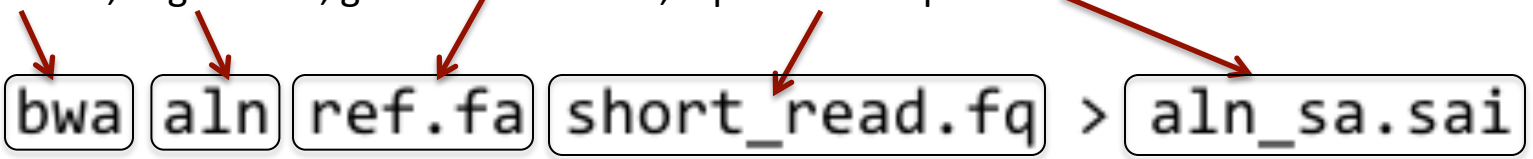
Align & Map

Command line

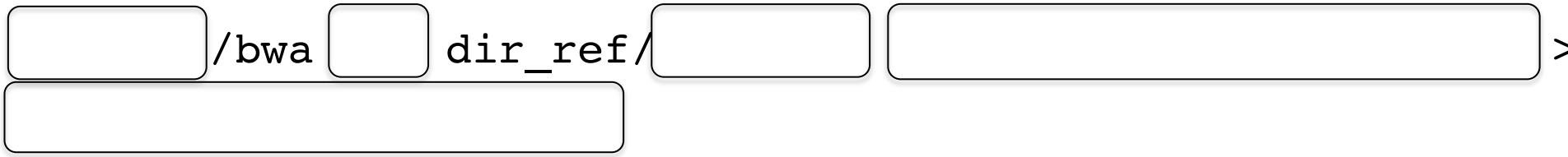
Two steps – 1 (has to be done for pe1 & pe2)

MANUAL

Run BWA, align mode, genome reference, input file output file



Based on the the above indicate which words go in which box



aln
2_sic_human001_pe1.fastq
dir_bwa
3_bwa_human001_pe1.sai
chr14.fa

BWA

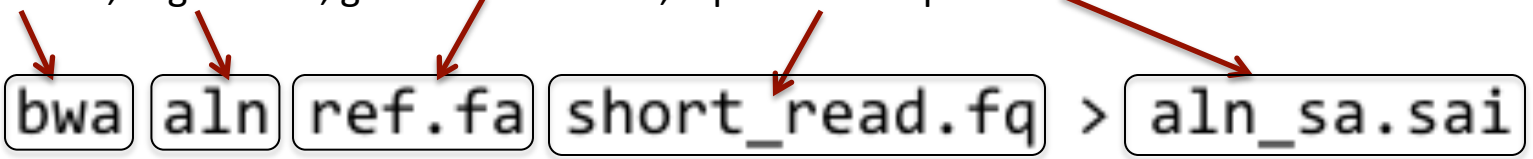
Align & Map

Command line

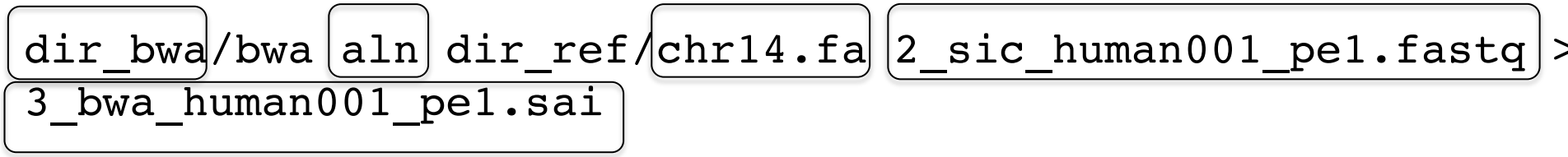
Two steps – 1 (has to be done for pe1 & pe2)

MANUAL

Run BWA, align mode, genome reference, input file output file



Based on the the above indicate which words go in which box



aln
2_sic_human001_pe1.fastq
dir_bwa
3_bwa_human001_pe1.sai
chr14.fa

BWA

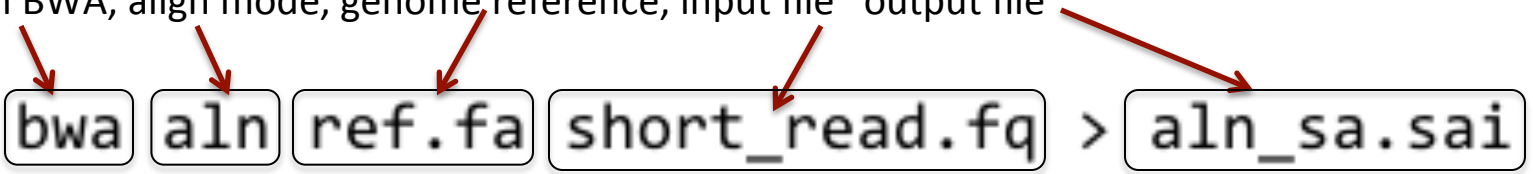
Align & Map

Command line

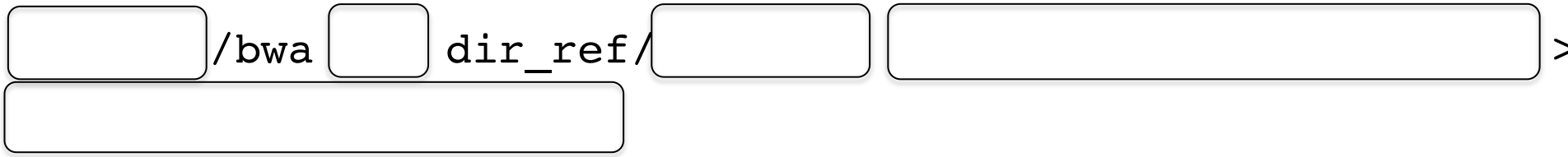
Two steps – 1 (has to be done for pe2 now)

MANUAL

Run BWA, align mode, genome reference, input file output file



Based on the the above indicate which words go in which box




```
aln
2_sic_human001_pe2.fastq
$dir_bwa
3_bwa_human001_pe2.sai
chr14.fa
```

BWA

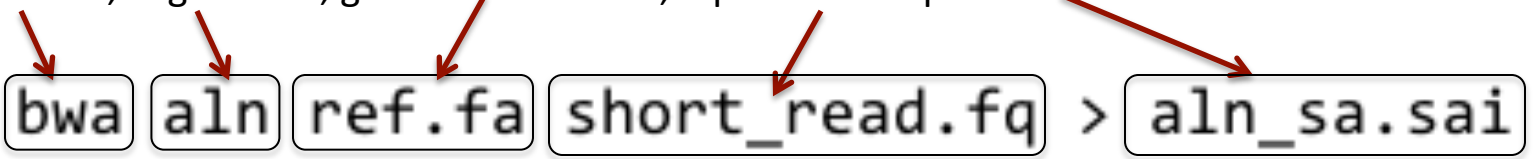
Align & Map

Command line

Two steps – 1 (has to be done for pe1 & pe2)

MANUAL

Run BWA, align mode, genome reference, input file output file



Based on the the above indicate which words go in which box

```
$dir_bwa/bwa aln dir_ref/chr14.fa 2_sic_human001_pe2.fastq >
3_bwa_human001_pe2.sai
```

```
aln
2_sic_human001_pe1.fastq
$dir_bwa
3_bwa_human001_pe1.sai
chr14.fa
```

BWA

Align & Map

Command line

Two steps -2

MANUAL

INPUT FILES
- sai files from step 3
- fastq files from step 2

run BWA, paired end mode, reference,

output

```
bwa sampe ref.fa aln_sa1.sai aln_sa2.sai read1.fq read2.fq > aln-pe.sam
```

Based on the the above indicate which words go in which box

```
$dir_bwa/bwa sampe $dir_ref/chr14.fa
```

>

```
2_sic_human001_pe1.fastq
2_sic_human001_pe2.fastq
3_bwa_human001_pe1.sai
3_bwa_human001_pe2.sai
4_bwa_human001_pe12.sam
```

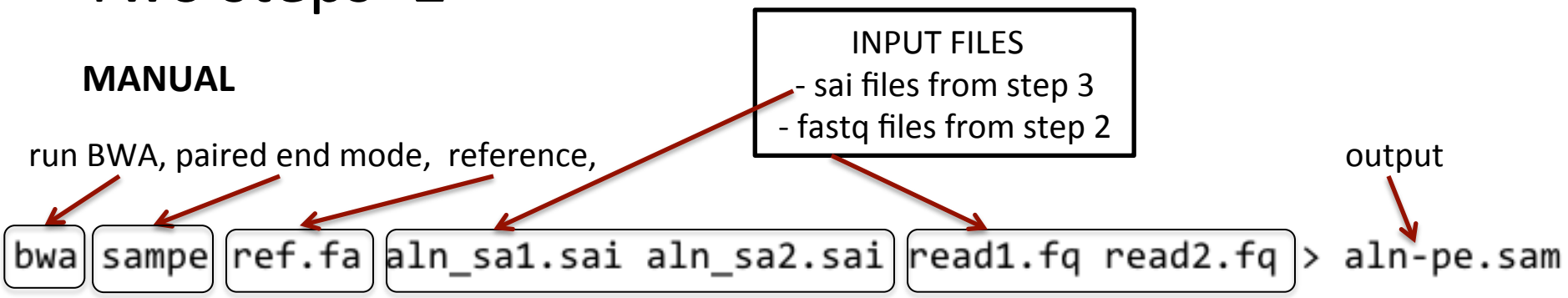
BWA

Align & Map

Command line

Two steps -2

MANUAL



Based on the the above indicate which words go in which box

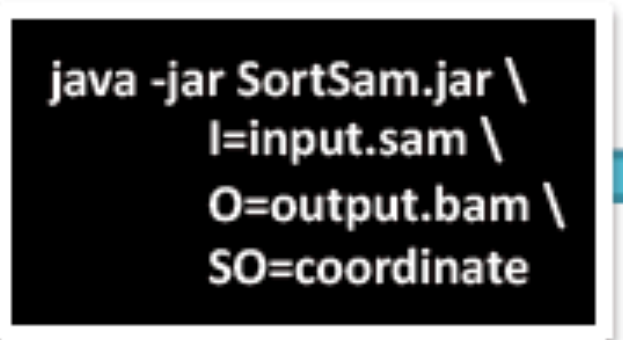
```
$dir_bwa/bwa sampe $dir_ref/chr14.fa 3_bwa_human001_pe1.sai
3_bwa_human001_pe2.sai 2_sic_human001_pe1.fastq
2_sic_human001_pe2.fastq > 4_bwa_human001_pe12.sam
```

```
2_sic_human001_pe1.fastq
2_sic_human001_pe2.fastq
3_bwa_human001_pe1.sai
3_bwa_human001_pe2.sai
4_bwa_human001_pe12.sam
```

Sam to Bam file

Lets convert our sam file to a bam file using picard tools

MANUAL



We also have to add: CREATE_INDEX=true

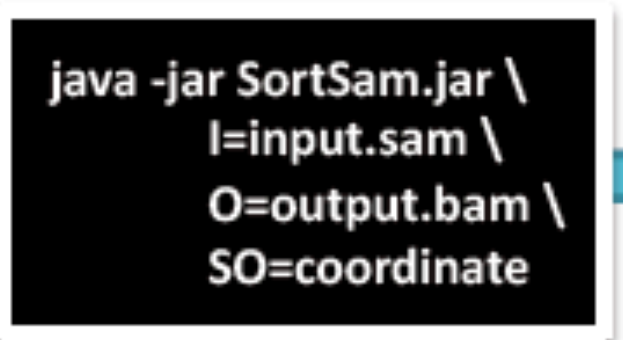
java -jar [ ]/SortSam.jar I=[ ]
O=[ ] SO=coordinate CREATE_INDEX=true

$dir_pic
5_bwasort_human001_pe12.bam
4_bwa_human001_pe12.sam

Sam to Bam file

Lets convert our sam file to a bam file using picard tools

MANUAL



We also have to add: CREATE_INDEX=true

```
java -jar $dir_pic/SortSam.jar I=4_bwa_pe12.sam
O=5_bwasort_pe12.bam SO=coordinate CREATE_INDEX=true
```

```
$dir_pic
5_bwasort_human001_pe12.bam
4_bwa_human001_pe12.sam
```

Filtering out poorly mapping reads

What if wanted to excluded any unmapped reads or reads with low mapping scores?

Filter out unmapped reads

Filter out reads if mapping quality < 30

```
samtools view –F 4 –q 30 –b mybamfile > myoutfile
```

```
        /samtools view –F 4 –q 30 –b

>
```

```
5_bwasort_human001_pe12.bam
$dir_sam
6_bwafil_human001_pe12.bam
```

Filtering out poorly mapping reads

What if wanted to excluded any unmapped reads or reads with low mapping scores?

Filter out unmapped reads

Filter out reads if mapping quality < 30

```
samtools view –F 4 –q 30 –b mybamfile > myoutfile
```

```
$dir_sam/samtools view –F 4 –q 30 –b 5_bwasort_human001_pe12.bam

> 6_bwafil_human001_pe12
```