

# disbamer.sh

```
$ bash disbamer.sh 43 | less -S
$ bash disbamer.sh 43
```

Was a read number supplied?

`##` : Number of arguments passed to bash script

We looking for one input only!

```
if [[ $# -ne 1 ]]; then
    echo messages
    exit 2
fi
```

Is samtools loaded?

Try samtools, output to null, and redirect error to stdout, if we get output then show message

```
> command -v samtools >/dev/null 2>&1
|| { echo >&2 "- Is samtools
loaded? "; exit 1; }
```

Extract Read from SAM file.

run samtools, use sed to get line requested, and put read in to var bamdata

```
> bamdata=$(samtools view sam.lnk |
sed "${1}q;d")
```

Get read data fields.

Use awk to get fields from bamdata. Bash 'read' puts output into variable : seqD

```
> read seqD <<< $(echo "$bamdata" | awk
'{print $10}')
# Repeat for flag region position
sequence cigar ...
```

Check sequence field.

Secondary alignments don't hold sequence

Sam data only has sequence in primary/supplementary mapped reads.

```
> if [ "$seqD" == "*" ]; then
    echo "No sequence data : secondary?"
    exit 1
fi
```

Cont...

# disbamer.sh

> bash disbamer.sh 43 | less -S

## Using CIGAR calculate length of Reference sequence required

awk script

Use awk script to calculate.  
Use bash 'read' to assign  
result. to variable, returns  
'\*' if error

```
> read seqlengthD <<< $(awk -v  
cigA="$cigarD"-f cigtoRefLen.awk)
```

## Obtain reference matching sequence for read

awk script

Send bash variables for  
'region, position and  
length' to the awk script;  
return sequence

```
> read seqrefD <<< $(awk -v  
regA="$regionD"-v posA="$positionD"-v  
lenA="$seqlengthD"-f getrefseq.awk  
ref.lnk)
```

## Display read alongside Reference with indels marked

awk script

Send bash variables for  
'cigar, read and reference  
sequences' to the awk  
script for display

```
> awk -v cigA="$cigarD"-v seqA="$seqD"-v  
refA="$seqrefD"-f viewread.awk
```

# cigtoRefLen.awk

Using CIGAR calculate length of  
Reference sequence required

```
read -r seqlengthD <<< "$(awk -v cigA="$cigarD" -f cigtoRefLen.awk)"
```

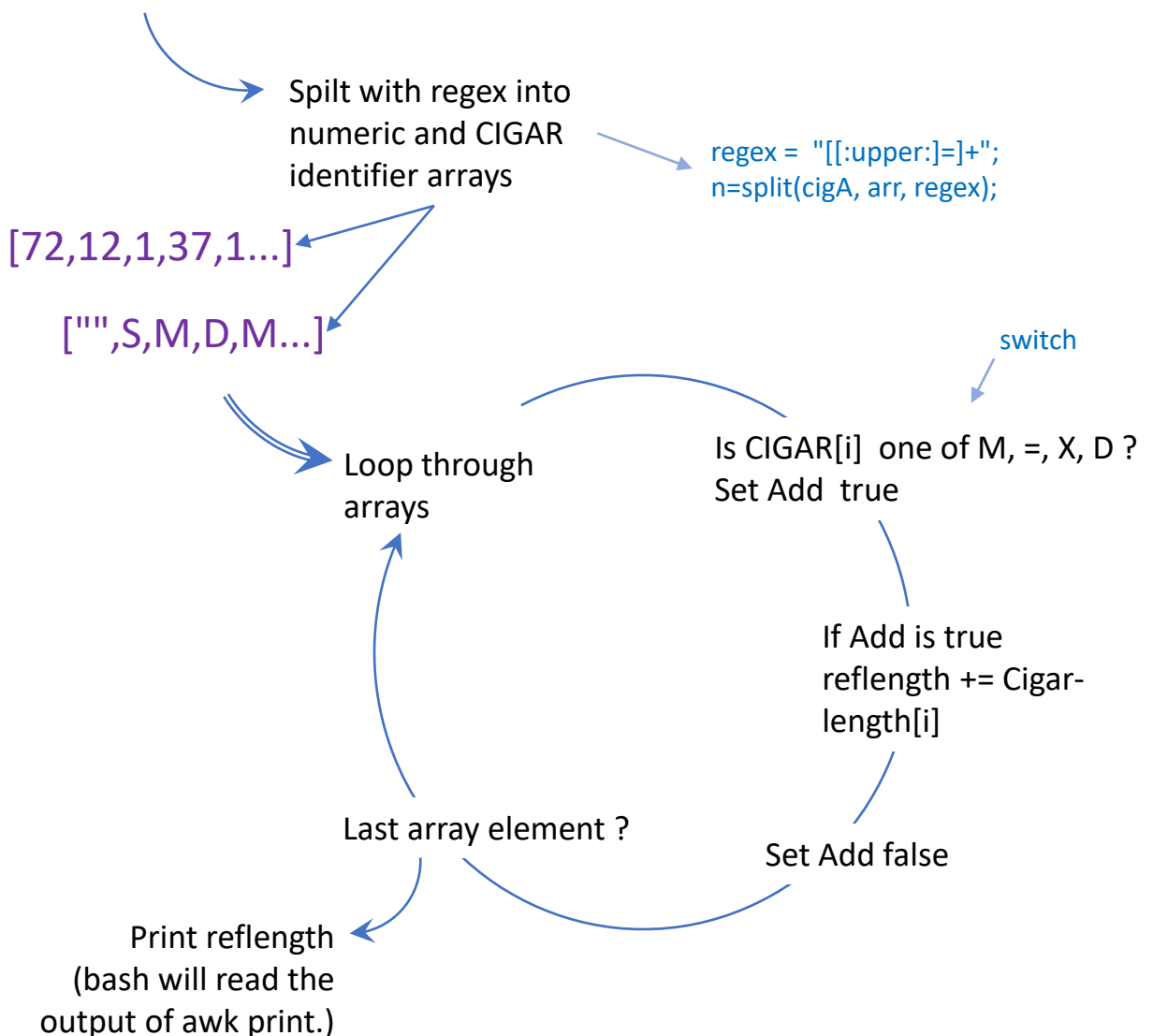
Input:

- Cigar String,

Output:

- Ref Length

72S12M1D37M1D10M2D2M3D14M1D4M1D23M2D1M1I24M692S



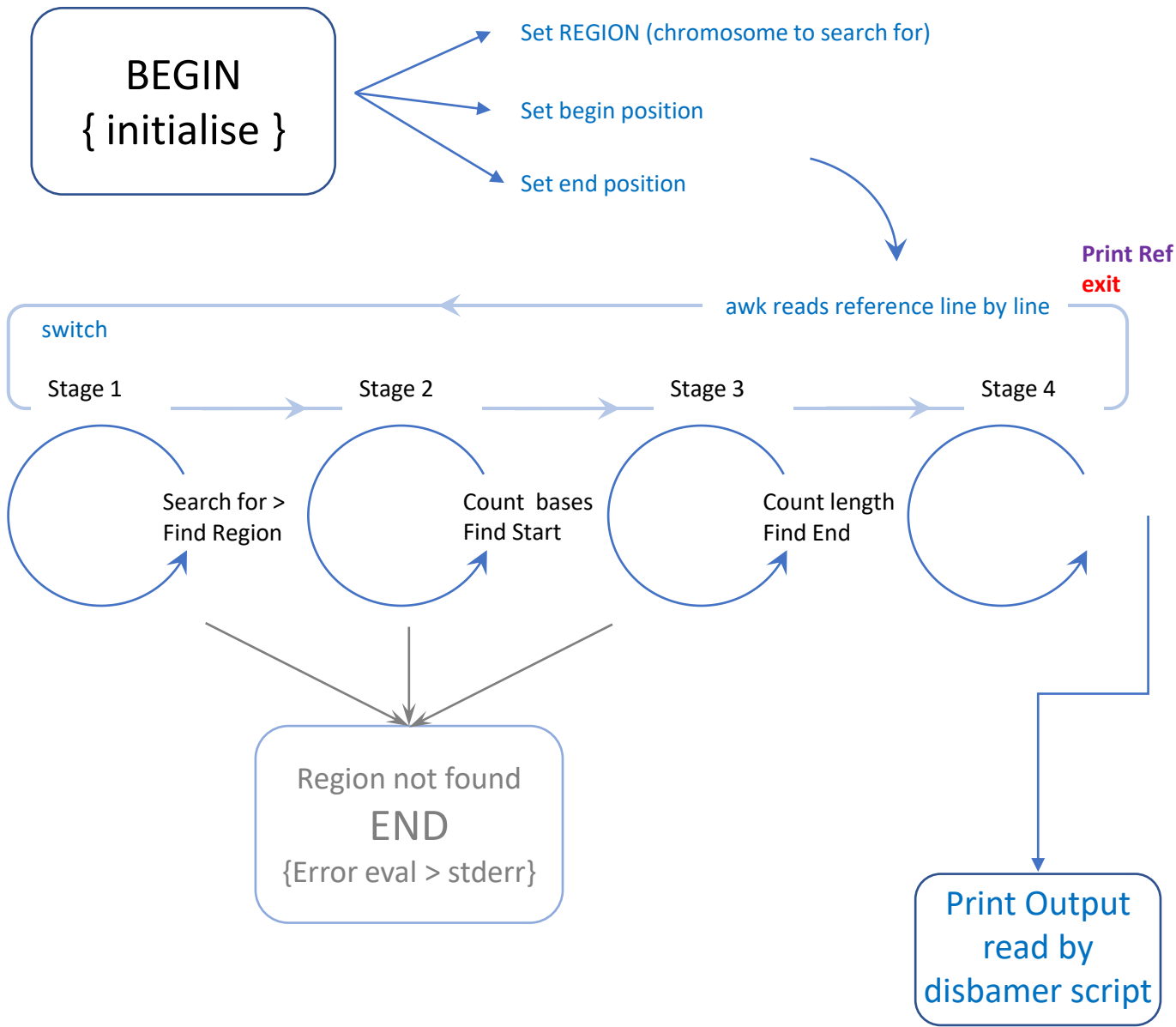
Note: All code is in a BEGIN block, and only operates on passed variable (via -v)

# getrefseq.awk

Obtain reference matching sequence for read

```
read -r seqrefD <<< $(awk -v regA="$regionD" -v posA="$positionD" -v lenA="$seqlengthD" -f getrefseq.awk ref.lnk)
```

- Input:
- Region
  - Start
  - length
- Output:
- Ref Sequence



Reference File format:

region

>CM000663.2 chr 1, GRCh38  
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNN  
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNN  
ttccAATACCTCTGAAAAAACTGatccaaa  
:  
>KI270706.1 contig chr 1, GRCh38  
GAATTCAGCTGAGAAGAACAGGCAAG  
TAATTTAAGATTTTCTCCCCCTACgtaatt  
ACTTGCTGTTTAAGGAACTAATTAAAC  
:  
end

start

# viewread.awk

Display read alongside Reference with indels marked

```
awk -v cigar="$cigarD" -v seqA="$seqD" -v refA="$seqrefD" -v outview="$nopipe_out" -f viewread.awk
```

Input:

- Cigar String,
- Read Sequence (**seq**) and
- Reference sequence
- nopipe\_out

5M1I3M2D7M2I6M

TCCCCCTACTAATTACGCTGTT

TCCCCTACAGTAATTACGGTCTT

Spilt Cigar String via  
regex into numeric and  
CIGAR identifier arrays

step through CIGAR arrays  
[CIGAR] [length]

switch  
CIGAR

M: Add matching sequence to results

=: Add matching sequence to results

X: Add matching sequence to results (mismatches)

D: Add cigar element length of dashes "-" to results

I: Add quoted sequence to results (these are not in Reference)

N: Add cigar element length of N's "N" to results

P: Add cigar element length of P's "P" to results

S: If first Cigar then set start of result sequence to end of clip.  
Otherwise increment over clip.

H: do nothing hard clip sequences are not in sam file.

Build a copy of the Read  
Sequence, showing inserts  
and deletions. Also gather  
statistics.

- Matches
- Deletions
- Insertions
- Clip

For [M, =, X, D ] Build  
position string.  
For I (inserts) build  
string with quotes.

Read is now:

TCCCC'C'TAC--TAATTAC'TT'GCTGTT

deletion

insertions

For each char in formatted read  
Format Reference sub sequence  
to align to our read.

Build an indicator string of  
deletions, mismatches and  
additions.

': add insert indicator '+' into reference and indicator for  
length of insert

:- reference remains as is, indicator string has dashes

Otherwise : reference  
remains as is, indicator string  
has 'x' if reference value  
does not match read.

If output is being piped "|" then output will  
print on a single line.  
(ie. for viewing with less -S)  
Otherwise will print 80 base pairs per line to  
terminal, automatically set by bash variable  
\$nopipe\_out.

Output:

READ	→	TCCCC'C'TAC--TAATTAC'TT'GCTGTTT
Reference	→	TCCCC'+TACAGTAATTAC'++'GGTCTTT
Indicator	→	.....'+'.--.....'++'.X.X.....
Position	→	[ .....   10 .....   20 .....   30