

```

#!/usr/bin/perl
open (FILE, "Input.fastq") || die "can't: $!";
open (OUTPUT1,>"Output.fastq") || die "can't: $!";
while(<FILE>)
{
$line = $_;
chomp($line);
if($line=~/^@DJDP.*/)
{
$seq=<FILE>;
chomp($seq);
$seq1=$seq;
chop($seq);
$junk=<FILE>;
chomp($junk);
# chop($junk);
$qual=<FILE>;
chomp($qual);
$qual1=$qual;
chop($qual);
$j=17;
$i=0;
@mismatch=();
@long=();

while($i < 7)
{
$j=17;
while($j <= length($seq)-$i-4)
{
@mismatch=0;
@bits=();
@q_val=();
push(@bits,substr($seq,$i,$j));
push(@bits,substr($seq,$j+$i,$j));
push(@q_val,substr($qual,$i,$j));
if( $i+$j+$j > length($seq) )
{
$off=length($seq)-$j-$i;
$adpt=substr($seq,$i,$off);
}
else
{
$adpt=$bits[0];
}
}
}
}

```

```

$match= $adpt ^ $bits[1];
while ($match =~ /^[^\0]/g)
{
    $mismatch++;

}

if ($mismatch > int((length$adpt)/10)){
    $j=$j+1;
}else{
    $mismatch[$i] = $mismatch;
    $long[$i] = $j;
    $j= length($seq)+1;
    if($mismatch eq 0) {
        #
        print OUTPUT
"$line\n$bits[0]\t\t$bits[1]\n$junk\n$q_val[0]\n";
        #
        print OUTPUT1 "$line\n$bits[0]\n$junk\n$q_val[0]\n";
        $i=7;
    }
}
}

$i++;

}

unless (@mismatch) {
#
print OUTPUT "$line\n$bits[0]\t\t$bits[1]\n$junk\n$q_val[0]\n";
print OUTPUT1 "$line\n$seq1\n$junk\n$qual1\n";
next;
}

foreach (@mismatch) {
    if ($_ eq "") {
        $_ = 100;
    }
}

@index = 0;
$minval = $mismatch[$index];
for ( 0 .. $#mismatch )
{
    if ( $minval > $mismatch[$_] )
    {

```

```
$index = $_;
$mismatch[$_] = $minval;
}
}

# if ($mismatch[$index] > 0) {
$bits[0]=substr($seq,$index,$long[$index]);
$q_val[0]=substr($qual,$index,$long[$index]);
# print OUTPUT "$line\n$bits[0]\t$bits[1]\n$junk\n$q_val[0]\n";
print OUTPUT1 "$line\n$bits[0]\n$junk\n$q_val[0]\n";
# }
}

exit;
```