Finishing Fosmid 455-J01 (Revision 2)

Alan Tseng
3/5/07
Bio4342W

Abstract

The main goal of Bio 4342 research in 2007 is to generate enough information on the *Drosophila mojavensis* dot chromosome for comparison with the same chromosome in *D. virilis* and *D. melanogaster*. To accomplish this goal, the class has been centered on completing high quality sequence (phred > 30) of fosmids from *D. mojavensis*, which will allow chromosome comparisons to uncover previously unknown domains, motifs, and other characteristics of the dot chromosome, as well as any differences that may span the *Drosophila* genus. In my report, I will describe the processes I have used to finish my fosmid (445-J01), from the initial assembly to the final resulting contig.

Initial Assembly Analysis

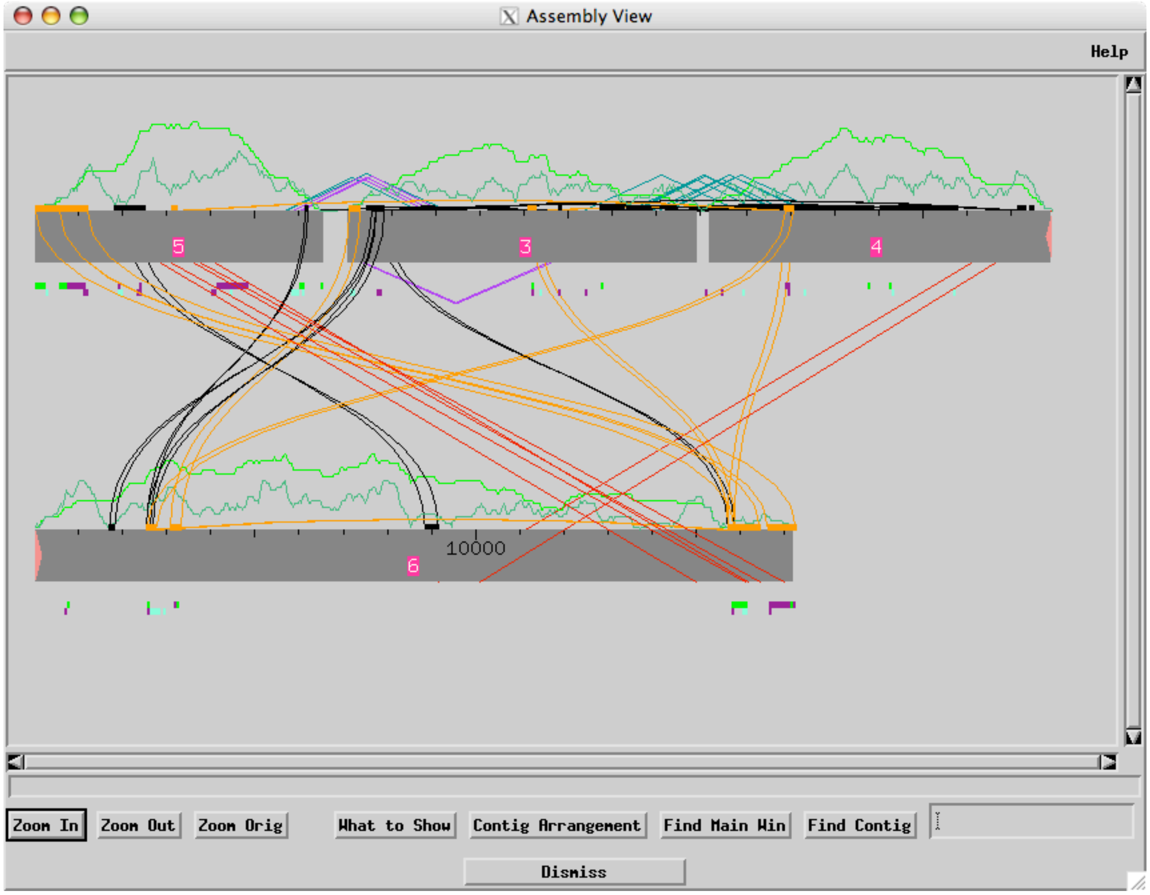


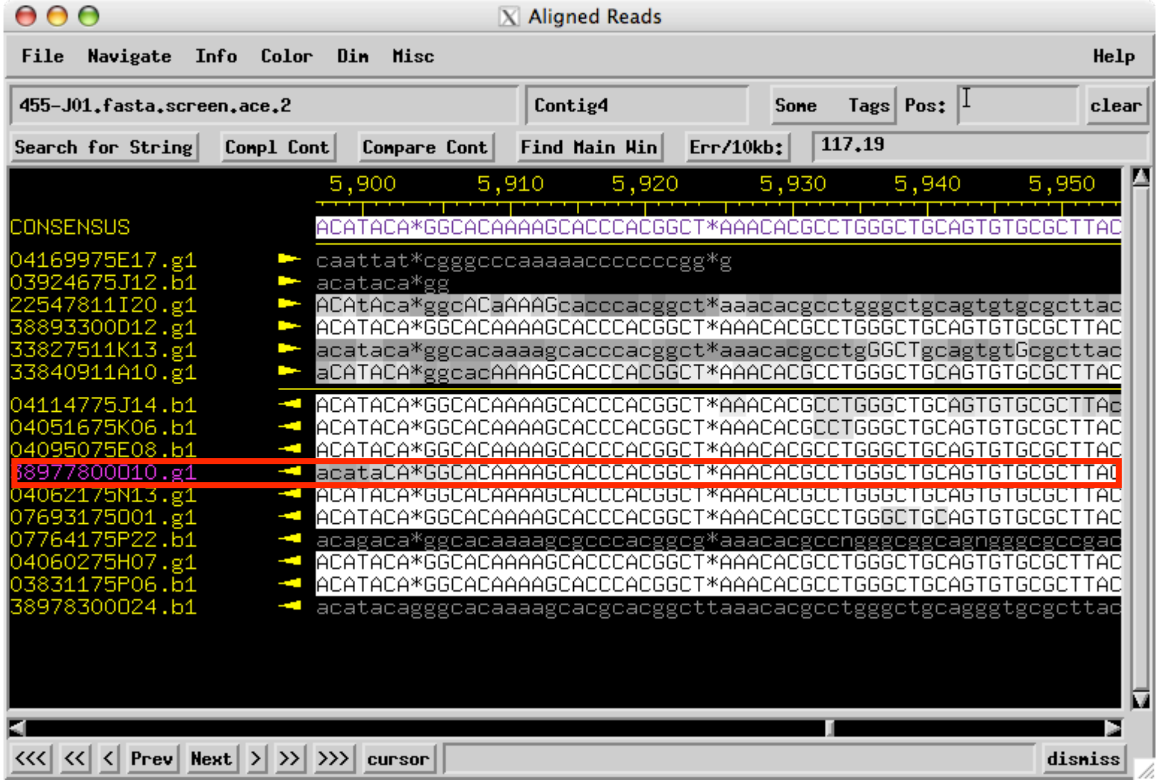
Figure 1: Initial Assembly View

My initial assembly consisted of 4 separate contigs, with contigs 6 and 4 (as they appear in Fig. 1) containing the end reads. Inconsistent forward/reverse pairs of sequence reads, indicated by red lines, occured between contigs 4 and 6, as well as contigs 6 and 5. Running Crossmatch displayed repeat structures identified by Consed throughout the fosmid, with orange lines signifying direct repeats and black lines indicating inverted repeats. Finally, low quality regions were prevalent in contig 3, and near the gaps between contigs.

My first concern when looking at this assembly view was the presence of repeat structures flanking the gaps between contigs 5 and 6 and contigs 5 and 3. Because some of these structures were larger than 1 kb in length, sequencing past these repeats into the gap would have been impossible. Thus it was difficult to discern whether these repeats overlapped or were actually located in tandem. A second concern I found in this initial assembly view was the forward and reverse pairs that were inconsistent because the two reads were located too far from each other. The inconsistent pairs that spanned contigs 5 and 6 had the possibility of being resolved if the gap-flanking repeats overlap, but the inconsistent pairs between contigs 6 and 4, which were the two end contigs, must have been due to misassembly.

Inconsistent Forward/Reverse Pairs

The first step I took was to resolve the inconsistent forward/reverse pairs across contig 4 and contig 6. To tackle this problem, I first looked to see whether any inconsistent reads were misassembled, which then led me to tear them out and attempt to reassemble them manually. When I saw that the reads aligned correctly, reassembling the reads in question became a trial and error procedure where I attempted to reassemble by using a particular sequence in *Search for String*. If no matches (with the exception of the match at its original location) were found, the read was left in its own contig. A read left in its own contig in this fashion was not assigned by random, but the decision was based on whether its absence would significantly reduce the quality of the data at its original position. Reads that were located in high quality regions were assigned according to this logic (Figure 2).


Aligned Reads
File Navigate Info Color Dim Misc Help
Some Tags Pos: clear
Search for String Compl Cont Compare Cont Find Main Win Err/10kb:
<<< << < Prev Next > >> >>> cursor dismiss

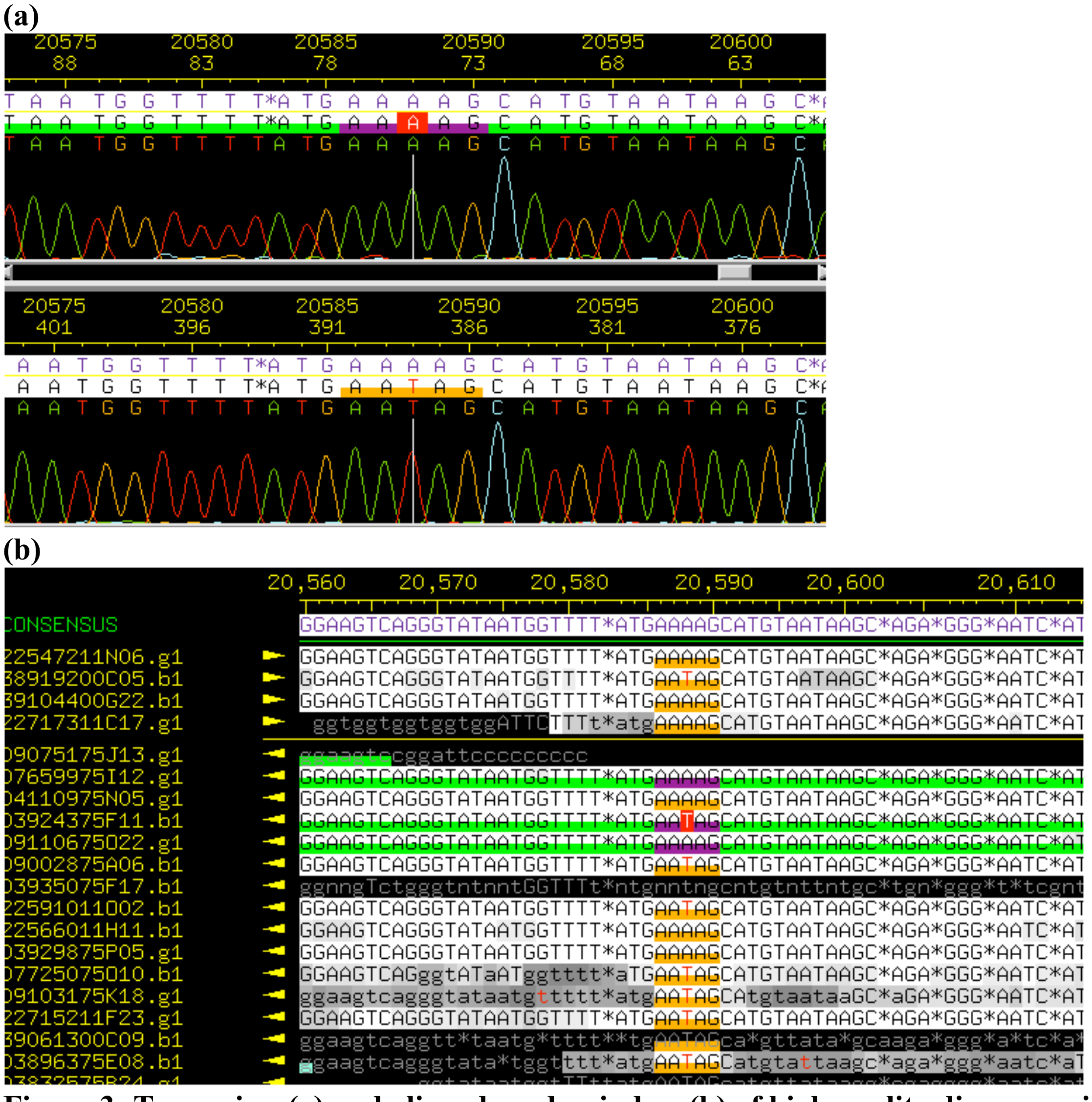
```
455-J01.fasta.screen.ace.2          Contig4
Err/10kb: 117.19

                      5,900      5,910      5,920      5,930      5,940      5,950
CONSENSUS             ACATACA*GGCACAAAAGCACCCACGGCT*AAACACGCCTGGGCTGCAGTGTGCGCTTAC
04169975E17.g1    ►   caattat*cgggcccaaaaaccccccgg*g
03924675J12.b1    ►   acataca*gg
22547811I20.g1    ►   ACAtAca*ggcACaAAAGcacccacggct*aaacacgcctgggctgcagtgtgcgcttac
38893300D12.g1    ►   ACATACA*GGCACAAAAGCACCCACGGCT*AAACACGCCTGGGCTGCAGTGTGCGCTTAC
33827511K13.g1    ►   acataca*ggcacaaaagcacccacggct*aaacacgcctgGGCTgcagtgtGcgcttac
33840911A10.g1    ►   aCATACA*ggcacAAAAGCACCCACGGCT*AAACACGCCTGGGCTGCAGTGTGCGCTTAC
04114775J14.b1    ◄   ACATACA*GGCACAAAAGCACCCACGGCT*AAACACGCCTGGGCTGCAGTGTGCGCTTAc
04051675K06.b1    ◄   ACATACA*GGCACAAAAGCACCCACGGCT*AAACACGCCTGGGCTGCAGTGTGCGCTTAC
04095075E08.b1    ◄   ACATACA*GGCACAAAAGCACCCACGGCT*AAACACGCCTGGGCTGCAGTGTGCGCTTAC
38977800O10.g1    ◄   acataCA*GGCACAAAAGCACCCACGGCT*AAACACGCCTGGGCTGCAGTGTGCGCTTAC
04062175N13.g1    ◄   ACATACA*GGCACAAAAGCACCCACGGCT*AAACACGCCTGGGCTGCAGTGTGCGCTTAC
07693175O01.g1    ◄   ACATACA*GGCACAAAAGCACCCACGGCT*AAACACGCCTGGGCTGCAGTGTGCGCTTAC
07764175P22.b1    ◄   acagaca*ggcacaaaagcgcccacggcg*aaacacgccngggcggcagngggcgccgac
04060275H07.g1    ◄   ACATACA*GGCACAAAAGCACCCACGGCT*AAACACGCCTGGGCTGCAGTGTGCGCTTAC
03831175P06.b1    ◄   ACATACA*GGCACAAAAGCACCCACGGCT*AAACACGCCTGGGCTGCAGTGTGCGCTTAC
38978300O24.b1    ◄   acatacagggcacaaaagcacgcacggcttaaacacgcctgggctgcagggtgcgcttac
```

Figure 2: Inconsistent read in a high coverage region

Out of the two inconsistent forward/reverse pairs between contigs 4 and 6, one remained unresolved and was removed from the assembly. As the read was pulled from a high coverage region, this did not affect the quality of my final assembly. With these inconsistent forward/reverse pairs resolved, I proceeded to look at high quality discrepancies within my fosmid.

High Quality Discrepancies

High quality discrepancies, such as the one shown in Figure 3, occurred in three places in my fosmid. My first method for dealing with these discrepancies was to tag the bases and tell phredPhrap not to overlap these regions. Re-running phredPhrap with these new settings then produced the assembly shown in Figure 4.


(a)
20575 88
20580 83
20585 78
20590 73
20595 68
20600 63
T A A T G G T T T T*A T G A A A A G C A T G T A A T A A G C*A
T A A T G G T T T T*A T G A A A A G C A T G T A A T A A G C*A
T A A T G G T T T T A T G A A A A G C A T G T A A T A A G C A
20575 401
20580 396
20585 391
20590 386
20595 381
20600 376
A A T G G T T T T*A T G A A A A G C A T G T A A T A A G C*A
A A T G G T T T T*A T G A A T A G C A T G T A A T A A G C*A
A A T G G T T T T A T G A A T A G C A T G T A A T A A G C A
(b)

```
                         20,560     20,570     20,580      20,590     20,600       20,610
CONSENSUS                GGAAGTCAGGGTATAATGGTTTT*ATGAAAAGCATGTAATAAGC*AGA*GGG*AATC*AT

22547211N06.g1       ►   GGAAGTCAGGGTATAATGGTTTT*ATGAAAAGCATGTAATAAGC*AGA*GGG*AATC*AT
38919200C05.b1       ►   GGAAGTCAGGGTATAATGGTTTT*ATGAATAGCATGTAATAAGC*AGA*GGG*AATC*AT
39104400G22.b1       ►   GGAAGTCAGGGTATAATGGTTTT*ATGAAAAGCATGTAATAAGC*AGA*GGG*AATC*AT
22717311C17.g1       ►    ggtggtggtggtggATTCTTTt*atgAAAAGCATGTAATAAGC*AGA*GGG*AATC*AT

D9075175J13.g1       ◄   ggaagtccggattccccccccc
D7659975I12.g1       ◄   GGAAGTCAGGGTATAATGGTTTT*ATGAAAAGCATGTAATAAGC*AGA*GGG*AATC*AT
D4110975N05.g1       ◄   GGAAGTCAGGGTATAATGGTTTT*ATGAAAAGCATGTAATAAGC*AGA*GGG*AATC*AT
D3924375F11.b1       ◄   GGAAGTCAGGGTATAATGGTTTT*ATGAATAGCATGTAATAAGC*AGA*GGG*AATC*AT
D9110675O22.g1       ◄   GGAAGTCAGGGTATAATGGTTTT*ATGAAAAGCATGTAATAAGC*AGA*GGG*AATC*AT
D9002875A06.b1       ◄   GGAAGTCAGGGTATAATGGTTTT*ATGAATAGCATGTAATAAGC*AGA*GGG*AATC*AT
D3935075F17.b1       ◄   ggnngTctgggtntnntGGTTTt*ntgnntngcntgtnttntgc*tgn*ggg*t*tcgnt
22591011O02.b1       ◄   GGAAGTCAGGGTATAATGGTTTT*ATGAATAGCATGTAATAAGC*AGA*GGG*AATC*AT
22566011H11.b1       ◄   GGAAGTCAGGGTATAATGGTTTT*ATGAAAAGCATGTAATAAGC*AGA*GGG*AATC*AT
D3929875P05.g1       ◄   GGAAGTCAGGGTATAATGGTTTT*ATGAAAAGCATGTAATAAGC*AGA*GGG*AATC*AT
D7725075O10.b1       ◄   GGAAGTCAGggtATaATggtttt*aTGAATAGCATGTAATAAGC*AGA*GGG*AATC*AT
D9103175K18.g1       ◄   ggaagtcagggtataatgttttt*atgAATAGCAtgtaataaGC*aGA*GGG*AATC*AT
22715211F23.g1       ◄   GGAAGTCAGGGTATAATGGTTTT*ATGAATAGCATGTAATAAGC*AGA*GGG*AATC*AT
39061300C09.b1       ◄   ggaagtcaggtt*taatg*tttt**tgAATAGca*gttata*gcaaga*ggg*a*tc*a*
D3896375E08.b1       ◄   ggaagtcagggtata*tggtttt*atgAATAGCatgtattaagc*aga*ggg*aatc*aT
```

Figure 3: Trace view (a) and aligned reads window (b) of high quality discrepancies

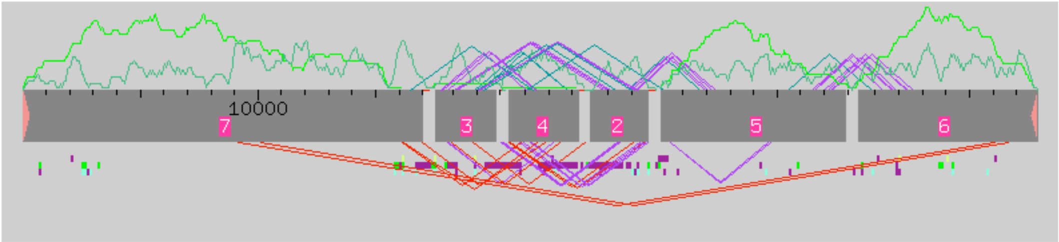
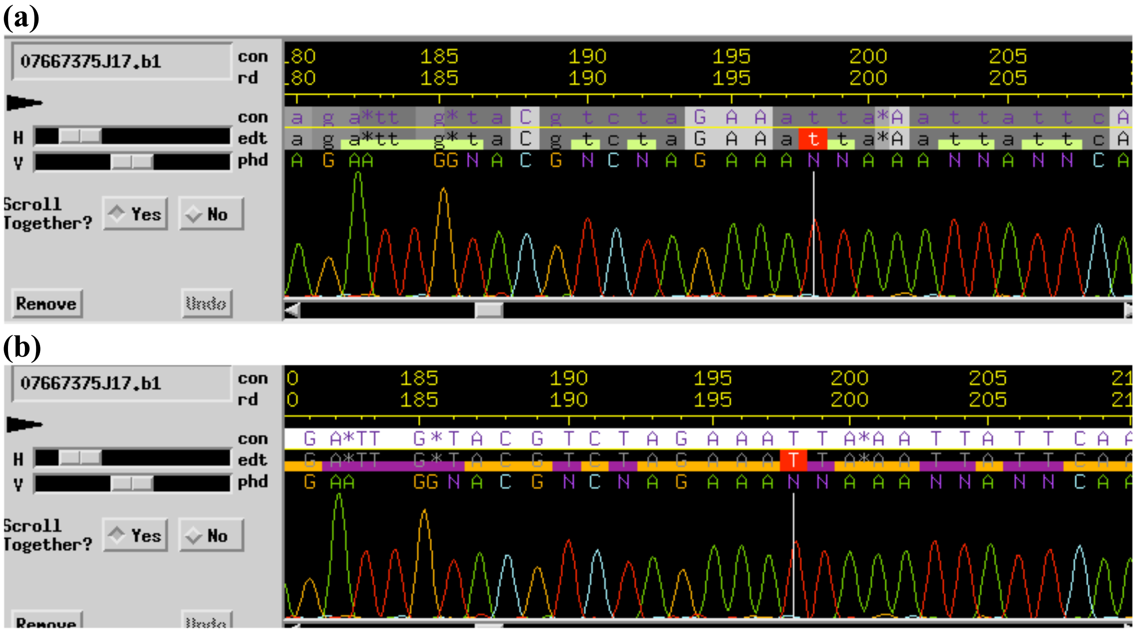


Figure 4: Re-running phredPhrap

With the highly inconsistent result from re-running phredPhrap, I (with the help and consultation of professional finishers) decided that the best explanation for these high quality discrepancies was single nucleotide polymorphisms. In addition, the 50/50 frequency of appearance of the two nucleotides also suggests a polymorphism, since this proportion gives higher confidence in the existence of two different bases at this location. On the other hand, a more uneven proportion of one polymorphism could lead to the conclusion that the appearance of the lower frequency base might be a miscall made by the base-calling program.

Editing Base Miscalls

In addition to looking at inconsistent forward reverse pairs and high quality discrepancies, I also edited base miscalls that allowed me to extend my gap-spanning reads a few hundred bases further out. These base miscalls were easy to fix, as the read was actually of high quality. Fixing the miscalls was simply a matter of striking in the correct base and changing the consensus to reflect the changes. This process is illustrated in Figure 5.


(a)

(b)

07667375J17.b1
con
rd
con
edt
phd
H
V
Scroll Together?
Yes
No
Remove
Undo
180 185 190 195 200 205
a g a*tt g* t a C g t c t a G A A a t t a*A a t t a t t c A
A G AA GG N A C G N C N A G A A A N N A A A N N A N N C A
185 190 195 200 205
G A*TT G*T A C G T C T A G A A A T T A*A A T T A T T C A
G AA GG N A C G N C N A G A A A N N A A A N N A N N C A

Figure 5: Base miscalls before editing (a) and after editing (b)

Round 1 reaction order

In the first round of reactions, I was focused on spanning the gaps between all of the contigs. Selection of the oligonucleotide primers involved choosing a suitable melting temperature (between 56-60°C) and a unique sequence. Table 1 lists the oligos used for the first round of reactions. Notice that only one oligo was called for the gap between contigs 5 and 6, since the presence of a gap-flanking repeat did not allow me to call an oligo on contig 5.

Table 1: Round 1 oligos

Oligo	Sequence	Dir	Chemistry	Problem	Result
1	gattcgtgccaacaaattta	->	Big Dye, 4:1	Gap 6-5	Not Added
2	ccatcttacatgggagtcttaaat	<-	Big Dye, 4:1	Gap 5-3	Added
3	acaggaagagctcctttaaaa	->	Big Dye, 4:1	Gap5-3	Not Added
4	ttatagctggagattttaactcct	<-	Big Dye, 4:1	Gap 3-4	Added
5	gcctggtaagctaactgtttct	->	Big Dye, 4:1	Gap 3-4	Added

While most of the oligos produced successful reads, not all of the chemistry choices were successful. Success of either the Big Dye or the 4:1 depended on the region sequenced, and rarely did both chemistries produce usable reads. However, with the incorporation of reads from three of these five reactions, I was able extend contigs 5, 4 and 3 and shorten the gaps. In addition, I was able to join contigs 4 and 3 to form contig 20 (Fig. 6).

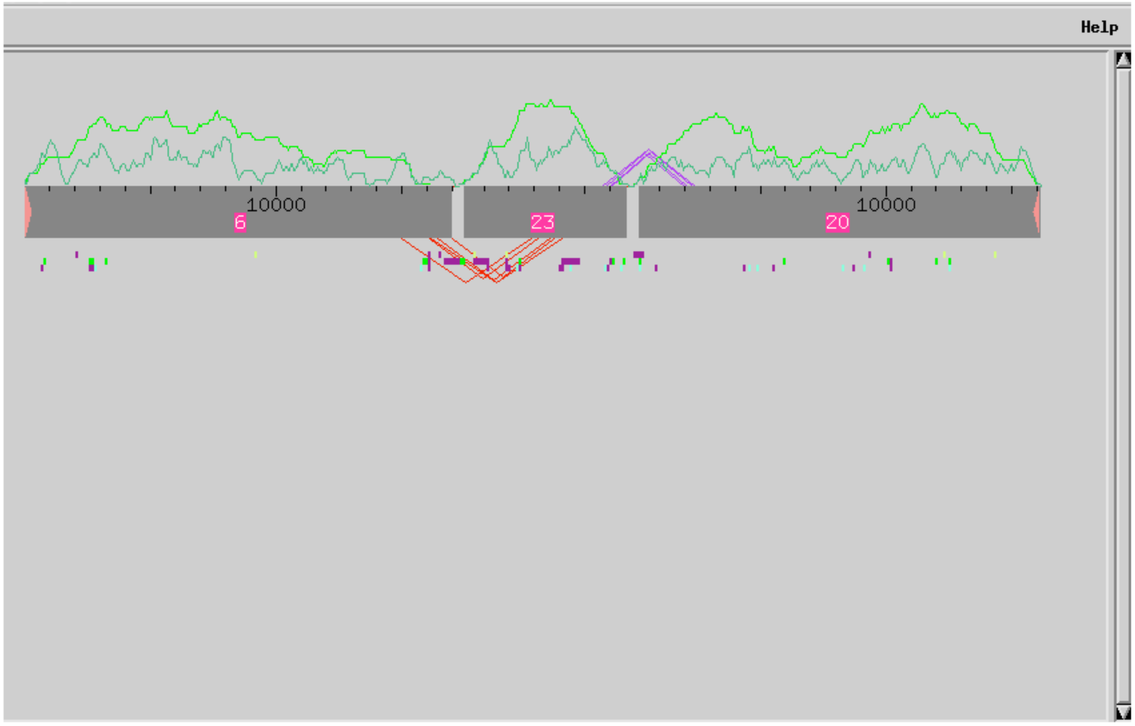


Figure 6: Result of round 1 reactions

Joining Contig 6 and 23

My next step in finishing my fosmid was to join contigs 6 and 23. Because my gap-spanning read for contig 6 did not succeed in generating sequence, the approach was to force join the two contigs and overlap the repeat sequences. This might not only resolve the inconsistent forward/reverse read pairs in this region, but it might also form a legitimate join.

To make this force join, I first analyzed the repeats on each of the two contigs. What I found was that contig 6 had two different repeats that were both duplicated on contig 23. Interestingly, these duplications were separated on each contig by two different unique sequences. In contig 6, the two repeats are separated by approximately 200 bases of unique sequence, while in contig 23, only 10 bases of unique sequence are present. In addition, the unique ten bases on contig 23 did not match any of the sequence in the 200 bases of unique sequence in contig 6, which meant that these two variants could not be explained by a single insertion/deletion event. Figure 7 shows these unique sequences in *Assembly View*.

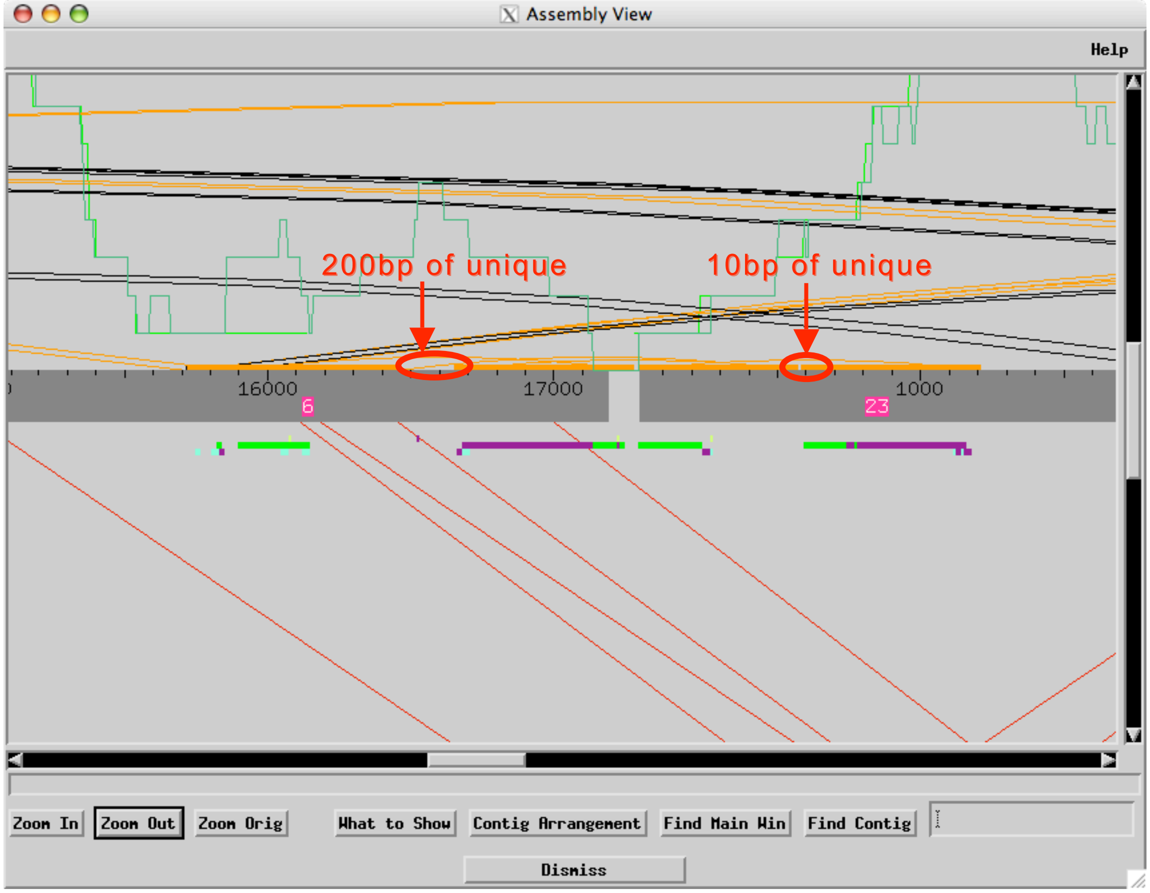
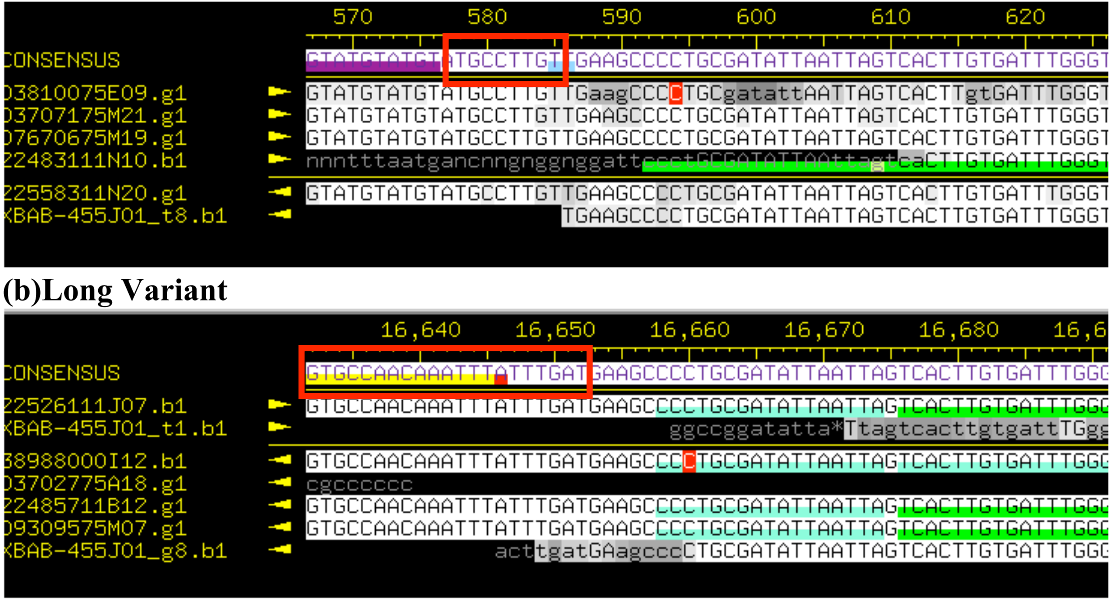


Figure 7: Unique sequences in contigs 6 and 23 (Assembly View)

Looking at the same areas in the *Aligned Reads Window* (Fig. 8), we see the two variations of unique sequences. The unique sequence (outlined in red) of the long variant is shown to extend off the window to the left, while the shorter variant is shown with two flanking repeat regions. One explanation for the presence of these two variants could be that the two variants are similar parts of the *Drosophila mojavensis* genome. Of the two, one of the variants belongs to my assembly, while the other variant could have been placed in this assembly by mistake. If this were true, then extraction of one of the variants would be necessary to join the two contigs together.

(a) Short Variant



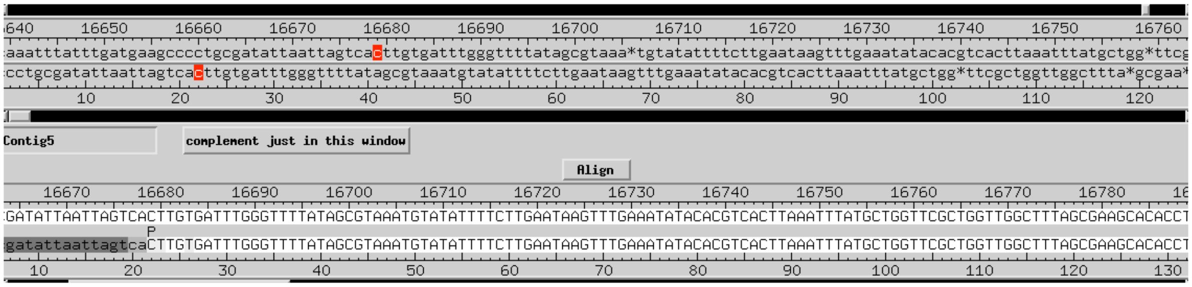
```
                          570       580       590       600       610       620
CONSENSUS                 GTATGTATGTATGCCTTGTTGAAGCCCCTGCGATATTAATTAGTCACTTGTGATTTGGGT
D3810075E09.g1         ►  GTATGTATGTATGCCTTGTTGaagCCCCTGCgatattAATTAGTCACTTgtGATTTGGGT
D3707175M21.g1         ►  GTATGTATGTATGCCTTGTTGAAGCCCCTGCGATATTAATTAGTCACTTGTGATTTGGGT
D7670675M19.g1         ►  GTATGTATGTATGCCTTGTTGAAGCCCCTGCGATATTAATTAGTCACTTGTGATTTGGGT
22483111N10.b1         ►  nnntttaatgancnngnggnggattccctGCGATATTAAttagtcaCTTGTGATTTGGGT
22558311N20.g1         ◄  GTATGTATGTATGCCTTGTTGAAGCCCCTGCGATATTAATTAGTCACTTGTGATTTGGGT
XBAB-455J01_t8.b1      ◄                       TGAAGCCCCTGCGATATTAATTAGTCACTTGTGATTTGGGT
```

(b)Long Variant

```
                               16,640    16,650    16,660    16,670    16,680    16,6
CONSENSUS                 GTGCCAACAAATTTATTTGATGAAGCCCCTGCGATATTAATTAGTCACTTGTGATTTGGG
22526111J07.b1         ►  GTGCCAACAAATTTATTTGATGAAGCCCCTGCGATATTAATTAGTCACTTGTGATTTGGG
XBAB-455J01_t1.b1      ►                               ggccggatatta*Ttagtcacttgtgatt TGgg
38988000I12.b1         ◄  GTGCCAACAAATTTATTTGATGAAGCCCCTGCGATATTAATTAGTCACTTGTGATTTGGG
D3702775A18.g1         ◄  cgcccccc
22485711B12.g1         ◄  GTGCCAACAAATTTATTTGATGAAGCCCCTGCGATATTAATTAGTCACTTGTGATTTGGG
D9309575M07.g1         ◄  GTGCCAACAAATTTATTTGATGAAGCCCCTGCGATATTAATTAGTCACTTGTGATTTGGG
XBAB-455J01_g8.b1      ◄                 acttgatGAagcccCTGCGATATTAATTAGTCACTTGTGATTTGGG
```

Figure 8: Unique Sequences in contigs 23 (a) and contig 6 (b) outlined in red (Aligned Reads Window)

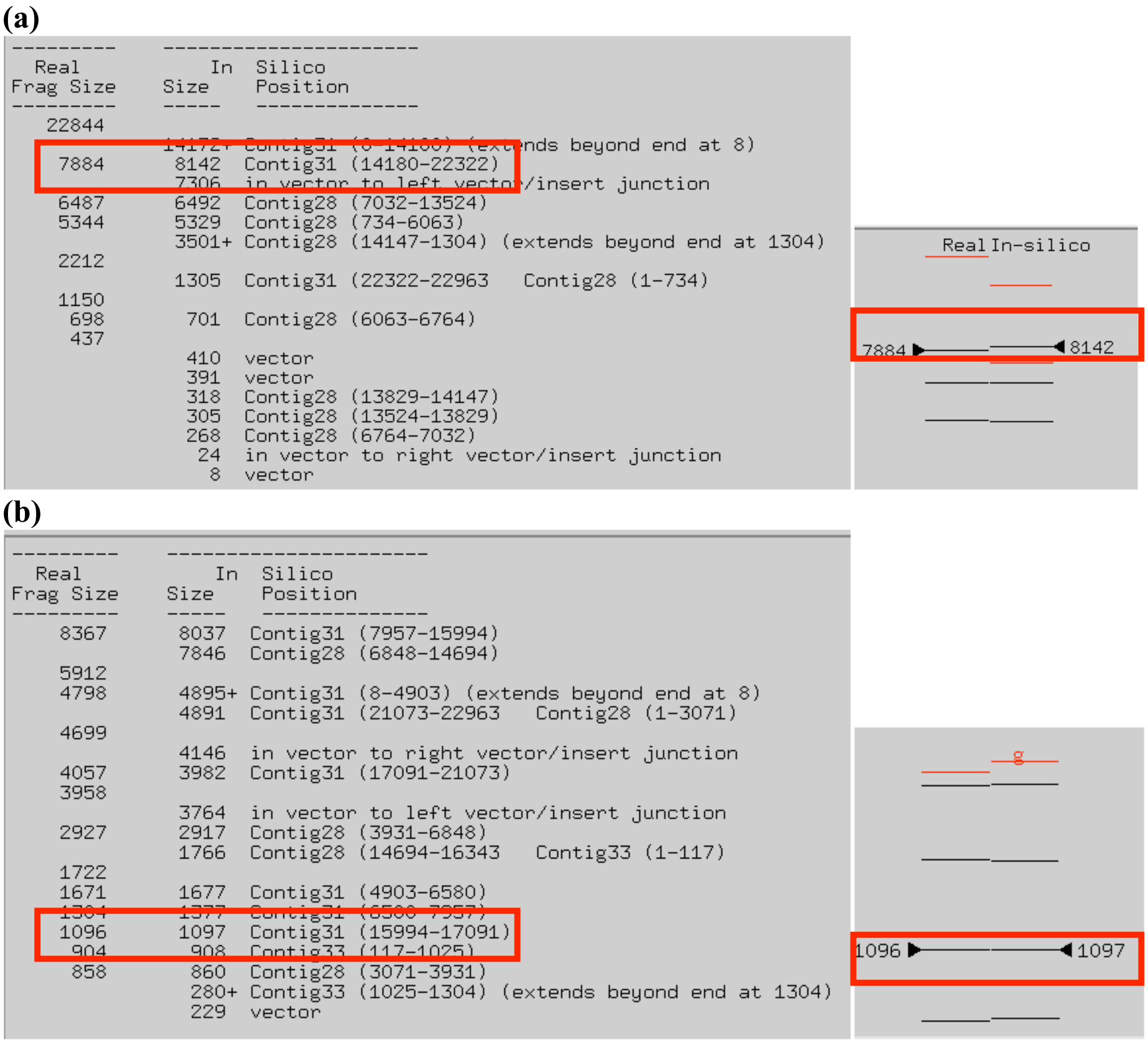
The decision was made to extract the shorter variant and place it into its own contig, as the convention is to always keep the longer variant that contains the most data. After the extraction, the resulting contigs were then joined at base position 16679 on contig 6 and 22 on contig 23 (Fig. 9), and *mini-assembly* was used to put all the reads with the short variant into one contig. A total of 6 reads contained the short variant.



```
640      16650     16660     16670     16680     16690     16700     16710     16720     16730     16740     16750     16760
aaatttatttgatgaagcccctgcgatattaattagtcacttgtgatttgggttttatagcgtaaa*tgtatatttcttgaataagtttgaaatatacacgtcacttaaatttatgctgg*ttcg
cctgcgatattaattagtcacttgtgatttgggttttatagcgtaaatgtatattttcttgaataagtttgaaatatacacgtcacttaaatttatgctgg*ttcgctggttggcttta*gcgaa*
        10        20        30        40        50        60        70        80        90       100       110       120

Contig5          complement just in this window

                                                  Align
    16670     16680     16690     16700     16710     16720     16730     16740     16750     16760     16770     16780     16
GATATTAATTAGTCACTTGTGATTTGGGTTTTATAGCGTAAATGTATATTTTCTTGAATAAGTTTGAAATATACACGTCACTTAAATTTATGCTGGTTCGCTGGTTGGCTTTAGCGAAGCACACCT
               P
gatattaattagtcaCTTGTGATTTGGGTTTTATAGCGTAAATGTATATTTTCTTGAATAAGTTTGAAATATACACGTCACTTAAATTTATGCTGGTTCGCTGGTTGGCTTTAGCGAAGCACACCT
   10        20        30        40        50        60        70        80        90       100       110       120       130
```

Figure 9: Force joining the two repeats

After joining these two contigs, it was necessary to determine whether or not the right join had been made. In the best-case scenario, if the repeats did overlap with each other correctly, then my join would be successful. On the other hand, if these repeat structures actually appeared in tandem, then the entire process would have resulted in a false join. To check the validity of the join, I checked the restriction digest data for my fosmid at the join region to see if the *in silico* data was in fact consistent with the actual gel image. The results of this analysis are shown in Figure 10. For this purpose, I compared the results from two restriction enzymes, *EcoRV* and *SacI*. To find the join region in the digest images, I looked for the digest fragments that contained base position 16679 on contig 31 (outlined in red). Looking at these regions, it was clear that my join region was indeed consistent with digest data. Therefore, I was confident in my join.



(a)

```
---------    ----------------------
  Real           In  Silico
Frag Size    Size    Position
---------    -----   --------------
   22844
             14172+ Contig31 (8-14180) (extends beyond end at 8)
    7884      8142  Contig31 (14180-22322)
              7306  in vector to left vector/insert junction
    6487      6492  Contig28 (7032-13524)
    5344      5329  Contig28 (734-6063)
              3501+ Contig28 (14147-1304) (extends beyond end at 1304)
    2212
              1305  Contig31 (22322-22963   Contig28 (1-734)
    1150
     698       701  Contig28 (6063-6764)
     437
               410  vector
               391  vector
               318  Contig28 (13829-14147)
               305  Contig28 (13524-13829)
               268  Contig28 (6764-7032)
                24  in vector to right vector/insert junction
                 8  vector
```

(b)

```
---------    ----------------------
  Real           In  Silico
Frag Size    Size    Position
---------    -----   --------------
    8367      8037  Contig31 (7957-15994)
              7846  Contig28 (6848-14694)
    5912
    4798      4895+ Contig31 (8-4903) (extends beyond end at 8)
              4891  Contig31 (21073-22963   Contig28 (1-3071)
    4699
              4146  in vector to right vector/insert junction
    4057      3982  Contig31 (17091-21073)
    3958
              3764  in vector to left vector/insert junction
    2927      2917  Contig28 (3931-6848)
              1766  Contig28 (14694-16343   Contig33 (1-117)
    1722
    1671      1677  Contig31 (4903-6580)
    1304      1377  Contig31 (6580-7957)
    1096      1097  Contig31 (15994-17091)
     904       908  Contig33 (117-1025)
     858       860  Contig28 (3071-3931)
               280+ Contig33 (1025-1304) (extends beyond end at 1304)
               229  vector
```

Figure 10: Digest data for the join with (a) SacI, and (b) EcoRV[1]

Now that I had ascertained the validity of this join, I created a comment tag on the short variant contig to notify the next finisher that the reads were taken out of the assembly on purpose, and that the longer variant was incorporated into the main contig. My assembly at this point is shown in Figure 11, with the short repeat variant placed into its own contig as contig 33.

[1] Gaps still existed in the assembly at this point, thus regions of inconsistency appeared in the restriction digest analysis

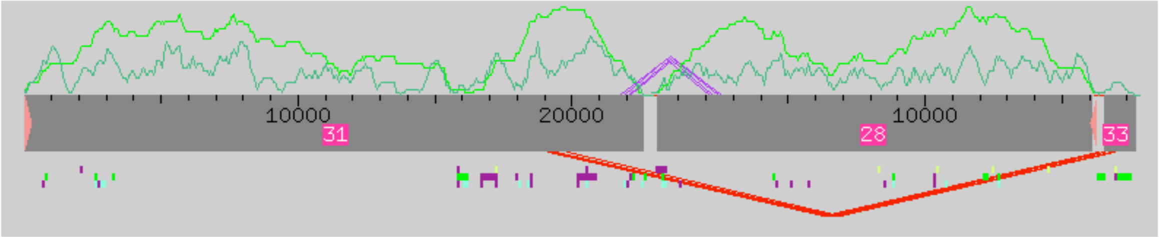


Figure 11: Assembly View after the force join

Round 2 reaction order

In round two of ordering sequencing reactions, I called oligos to try and resolve the gap between contig 31 and contig 28 (as they are numbered in Figure 9). In addition, I called numerous reactions to try to resolve some of the single stranded and single chemistry regions that still existed in my assembly. A combination of dye chemistries was chosen in order to produce the maximum number of successful reads. Table 2 outlines the reactions called in round 2 of reaction orders.

Table 2: Round 2 oligos

Oligo	Sequence	Dir	Chemistry	Problem	Result
7	ttgtgtaagctgaaagacgaaagat	->	Big Dye, 4:1	LQ region	Added
8	caggccccgttaccacac	<-	Big Dye, 4:1	LQ region	Added
3	acaggaagagctcctttaaaa	->	Big Dye, 4:1	Gap 31-28	Added
9	gaatttatttgttgaataatttaat	<-	All 3	Gap 31-28	Not Added
10	gggagtagagcttaaaagagtacag	->	Big Dye, 4:1	Gap 31-28	Not Added

Using these new reads, I was able to enhance a few low quality regions that needed more coverage. However, I was still unable to close the gap between contigs 31 and 28 due to low quality reads.

Round 3 reaction order and comparison with Autofinish

In my last round of calling reactions, I tried to call all the oligos I could in order to resolve all low quality regions in my contig. In addition, I called again some previous gap spanning oligos with different chemistries in a last attempt at closing the gap between contigs 31 and 28. These reactions are tabulated in Table 3.

Table 3: Round 3 oligos

Oligo	Sequence	Dir	Chemistry	Problem	Result
3	acaggaagagctcctttaaaa	->	Big Dye, 4:1	Gap 31-28	Added
9	gaatttatttgttgaataatttaat	<-	All 3	Gap 31-28	Added
10	gggagtagagcttaaaagagtacag	->	Big Dye, 4:1	Gap 31-28	Not Added
11	tgcgatttatactatttccgttt	->	Big Dye, 4:1	LQ region	Added
12	ggaagaaaattcaatttggtgt	<-	Big Dye, 4:1	LQ region	Added
13	tgaatgaaattatgaatatgaatcg	->	Big Dye, 4:1	LQ region	Added
14	acataatcgttttcaaatccc	<-	Big Dye, 4:1	LQ region	Added

Almost all of my sequencing reactions in this final round worked, but unfortunately, most of them did not reach as far as I would have liked. In the end, I was still left with some regions that were single stranded or had only one chemistry. When I compared all of my primer calls, starting from round 1 to round 3, to the calls made by *Autofinish*, I found that most of my calls agreed with *Autofinish* calls. Note that the *Autofinish* calls were made using the initial assembly. These calls are recorded in Table 4.

Table 4: Autofinish calls

Oligo	Sequence	Contig loc. and dir.	Problem
1	Cgctgtagtgtgggca	3 <-	Gap
2	Cgtgtttatgcgtgggt	3 <-	Gap
3	Ggcctcagcattaccaat	3 ->	Gap
4	Ggatacgggtctgttatcg	4 <-	Gap
5	agttaagttacgagtaacctcctct	4 ->	Gap
6	gcgctgtaattacgaacatt	5 <-	Gap
7	Tgaaccagcgctcagtag	5 <-	Gap
8	gctgaaagagcgattaccc	5 ->	Gap
9	ttccaataattctggtttactctta	6 <-	Gap
10	attggctctagcagtaattatttta	6 ->	Gap
11	Ggcggttcgctcagata	6 ->	Gap
12	Ccttgctgttcccacga	6 ->	Gap

Most of the *Autofinish* calls that were made were gap-spanning oligos in contigs 3, 4 and 5. The oligos on contigs 3 and 4 agreed well with my calls, but due to modification to contigs 5 and 6 resulting from my force join, the oligos that were called at those locations did not apply to my assembly. In addition, oligo calls made in contig 6 to correct single stranded regions were unnecessary after the force join of the repeat sequences provided data for both strands.

Joining Contig 31 and 28

Referring back to Fig. 11, I found that even though I attempted in all three rounds of reaction calling to call oligos that might close the gap between contigs 31 and 28, almost all of the reads were of low quality or did not reach far enough to form a join. The main reason why this gap was so hard to sequence was because an inverted repeat spanned the gap. Due to this inverted repeat, my oligos annealed to two locations during the reaction. As my sequencing reactions proceeded across the gap, the oligos would anneal again to the complimentary strand and begin a sequence reaction for a different sequence from that second location. What resulted was a double sequence in which two peaks occurred at each base call. An example of this phenomenon can be seen in Figure 12.

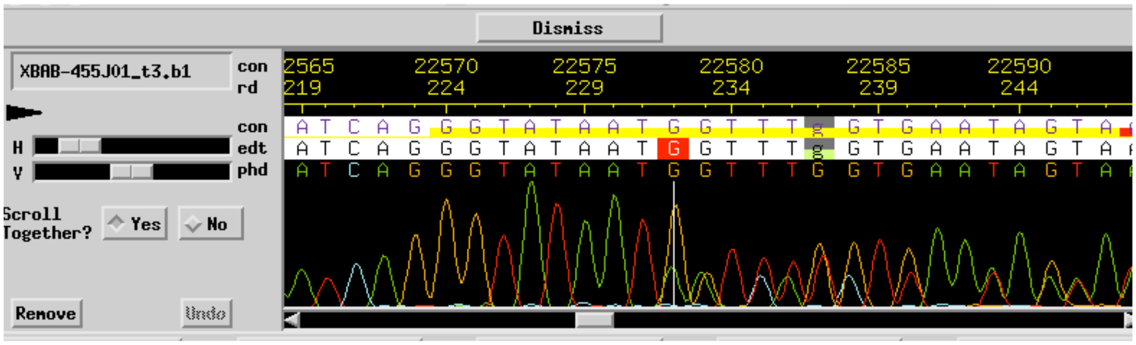

Dismiss
XBAB-455J01_t3.b1
con
rd
con
edt
phd
H
V
Scroll Together?
Yes
No
Remove
Undo
2565 22570 22575 22580 22585 22590
219 224 229 234 239 244
A T C A G G G T A T A A T G G T T T g G T G A A T A G T A
A T C A G G G T A T A A T G G T T T g G T G A A T A G T A
A T C A G G G T A T A A T G G T T T G G T G A A T A G T A

Figure 12: Double sequence read

Removing this read from the assembly during the final stages of the project would have resulted in loss of valuable information. Therefore, I kept reliable parts of the read but was forced to "n out" the areas of double sequence, which could not be trusted since correct base calls can not be distinguished in double peak regions. This process of changing all the low quality base calls to n's ensured that the double sequences would not affect the consensus sequence. A trace view of this process is shown in Figure 13.

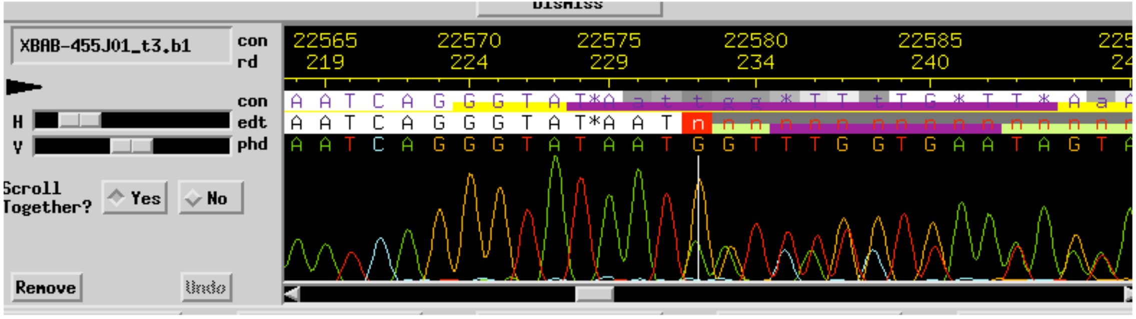
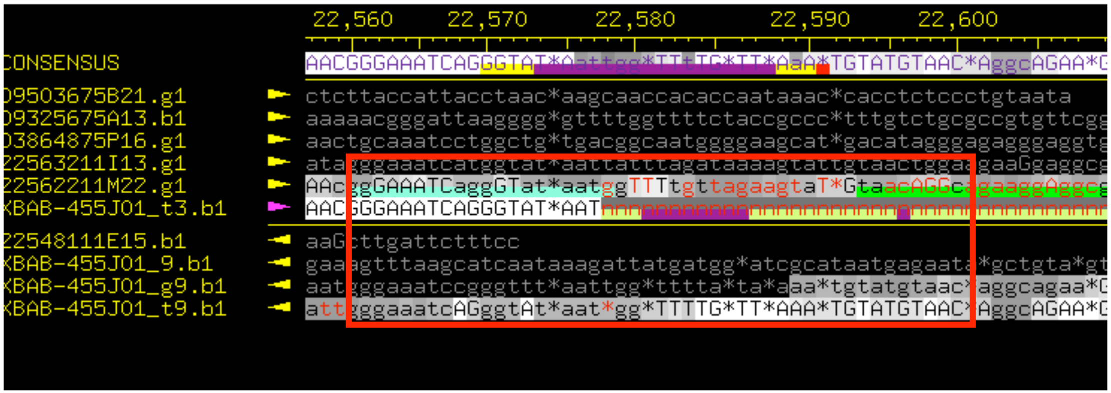

Dismiss
XBAB-455J01_t3.b1
con
rd
con
edt
phd
H
V
Scroll Together?
Yes
No
Remove
Undo
22565 22570 22575 22580 22585 22
219 224 229 234 240 24
A A T C A G G G T A T*A a t t g g * T T t T G * T T * A a A
A A T C A G G G T A T*A A T n n n n n n n n n n n n n n n n
A A T C A G G G T A T A A T G G T T T G G T G A A T A G T A

Figure 13: Correcting double peaked sequences

While this correction process drastically reduced my read lengths for the gap region, I was fortunate enough to come across a low quality sequence using *Search for String* that could be used for a forced join between the two low quality ends(Fig. 14).



```
                 22,560    22,570    22,580     22,590    22,600
CONSENSUS        AACGGGAAATCAGGGTAT*Aattgg*TTtTG*TT*AaA*TGTATGTAAC*AggcAGAA*G
09503675B21.g1   ctcttaccattacctaac*aagcaaccacaccaataaac*cacctctccctgtaata
09325675A13.b1   aaaaacgggattaagggg*gttttggttttctaccgccc*tttgtctgcgccgtgttcgg
03864875P16.g1   aactgcaaatcctggctg*tgacggcaatggggaagcat*gacatagggagagggaggtg
22563211I13.g1   atagggaaatcatggtat*aattatttagataaaagtattgtaactggaagaaGgaggcg
22562211M22.g1   AAcggGAAATCaggGTat*aatggTTTtgttagaagtaT*GtaacAGGcagaaggAggcg
XBAB-455J01_t3.b1 AACGGGAAATCAGGGTAT*AATnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnnn
22548111E15.b1   aaGcttgattctttcc
XBAB-455J01_9.b1 gaaagtttaagcatcaataaagattatgatgg*atcgcataatgagaata*gctgta*gt
XBAB-455J01_g9.b1 aatgggaaatccgggttt*aattgg*tttta*ta*aaa*tgtatgtaac*aggcagaa*G
XBAB-455J01_t9.b1 attgggaaatcAGggtAt*aat*gg*TTTTG*TT*AAA*TGTATGTAAC*AggcAGAA*G
```

Figure 14: Low quality join outlined in red[2]

[2] Note the 5' and 3' ends of this join is still discrepant and low quality; this can be rectified with addition of new high quality reads

In Figure 14, note that the top strands going right originate from contig 31, while the bottom strands are from contig 28. The red outline indicates the area of matching bases. Note that if the double sequence reads were to have been high quality, this join would have been a high quality join. Unfortunately, due to time constraints, the quality of the join region could not be improved. With this final join, my fosmid was now assembled into a single contig. This contig is shown Figure 15.

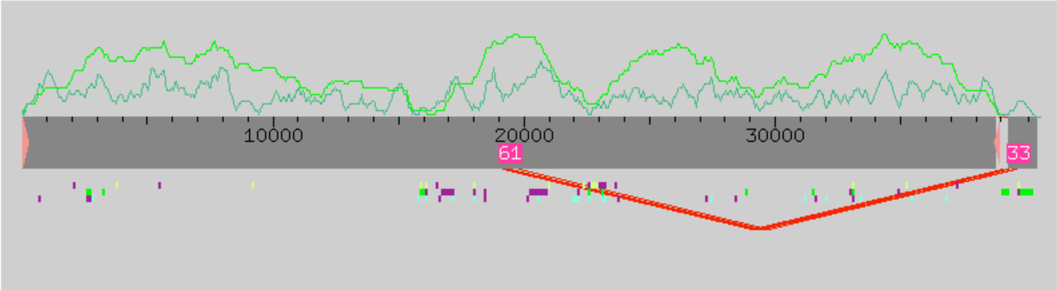


Figure 15: Final Assembly View

Digest Analysis

Once I completed my final join, I needed to support my contig assembly with restriction digest data, especially since the join I just made was a low quality join.

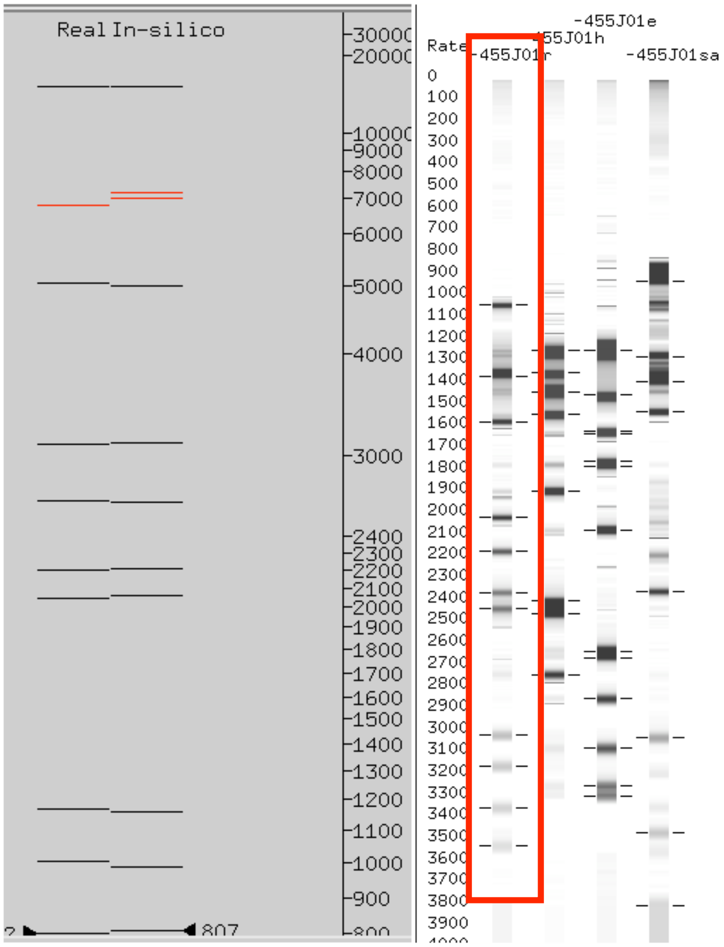
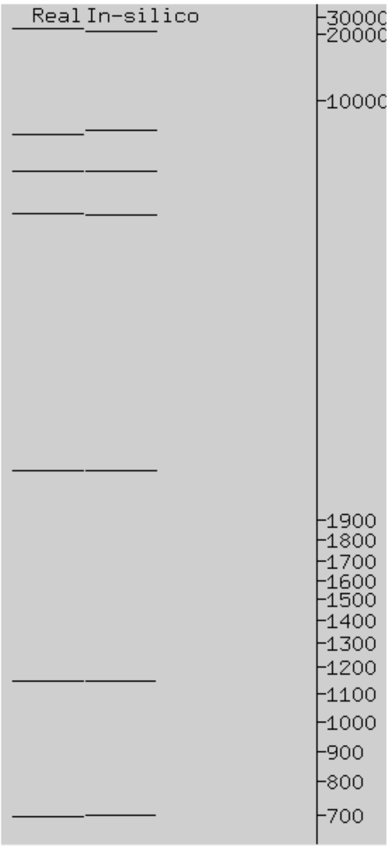


Figure 16: EcoRI Digest



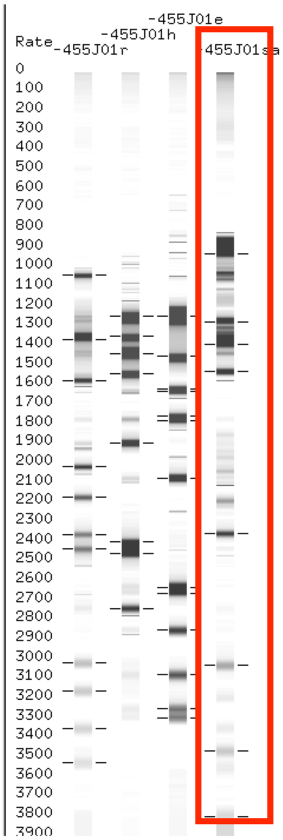


Figure 17: SacI Digest

From these results, it can be seen that the *in silico* analysis of my final contig is quite consistent with the actual gel images. The only discrepancy that can be seen is in the second band of the EcoRI digest. However, this minor inconsistency can be explained by looking at the actual gel image, which shows the second band as a thick band that could most likely be a doublet. Thus, if we were to suppose that the band calling program made a miscall, then our *in silico* data would be consistent with the gel data.

Conclusions:

In the end, I was able to reach the goal of producing one single contig for my fosmid. The reads that I called in the three rounds of sequencing reaction calling were consistent with the *Autofinish* calls made in the initial assembly. Final checking also revealed no single nucleotide runs in the assembly. However, due to time constraints, some problem regions have not been resolved. These regions include a low quality join region, three single stranded or single chemistry regions, and the high quality polymorphisms represented by contig 33. All of these regions have been tagged for further investigation by the next finisher.