Sonal Singhal
24 March 2006
Bio 4342—Finishing Paper: Final Draft

Finishing the Dot Chromosome of *D. virilis*: 99M21

DNA is often portrayed as a unidimensional object; most pictures of DNA show it as a linear form. However, in the cell, DNA is packaged tightly with proteins to form three-dimensional objects called chromosomes. The mode and mechanism of this condensation is still relatively unknown, but it is likely important. Research has suggested levels of condensation are correlated strongly to levels of gene expression. To better understand how DNA becomes a chromosome, our class is sequencing the fourth chromosome (or dot chromosome) of *Drosophila virilis*. Previous classes have already sequenced 2/3 of this chromosome. Upon completion, this sequence will be compared to the already-sequenced *Drosophila melanogaster* dot chromosome. The *D. melanogaster* dot chromosome is highly heterochromatic—i.e., tightly packed—yet initial analysis of its sequence suggests it has normal gene density. In contrast, the dot chromosome of *D. virilis* is euchromatic, or more loosely packed and presumably more permissive for transcription. By sequencing and annotating both dot chromosomes to high fidelity, we hope a comparative approach can help us determine how gene distribution and sequence organization contribute to heterochromatin formation. My contribution to the project is finishing a fosmid (99M21) containing approximately ~36 Kb of *D. virilis* sequence. This fosmid presented several challenges, which were solved by a variety of strategies. At completion, this fosmid still has two problem regions, but they are well-defined and likely to be resolved easily following additional read calls.

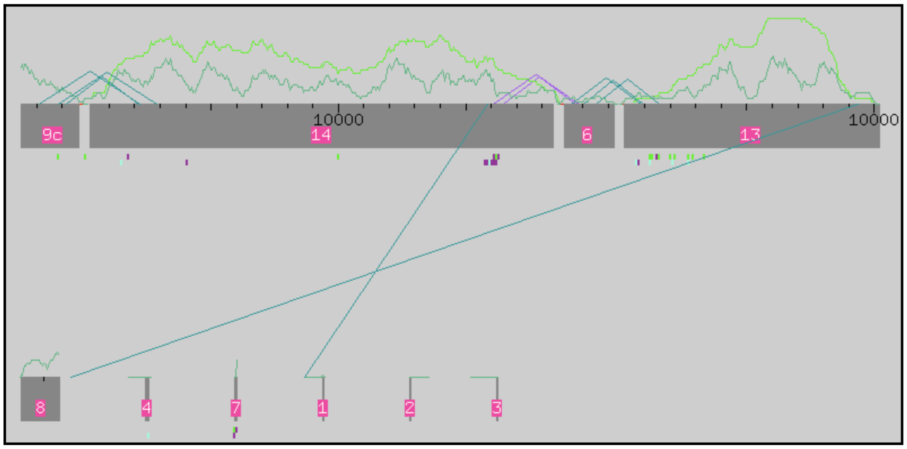


Figure 1: The initial assembly of 800 reads.

The sequences for my fosmid are derived from approximately 700 reads generated by the Genome Sequencing Center (GSC) using the normal pipeline and 96 reads generated by me. Using Phred and Phrap, the sequences are base-called and then assembled into contigs. As seen in Consed's Assembly View, the initial assembly consists of five major contigs separated by four gaps (Figure 1). Together, the gaps represent approximately 1 Kb of missing sequence; this estimate is based on Assembly View. My first objective was to determine the clone ends so that I could identify the orientation of the contigs. This year the GSC used a new vector to carry our fosmid subcloned fragments, and unfortunately this vector sequence had not been entered into the Phred/Phrap program. As such, the clone ends are misidentified. By looking for the "GATC" palindromic sequence that marks most vector ends, I was able to identify both of the clone ends correctly and thus determine the orientation of contigs in the assembly (Fig. 2).

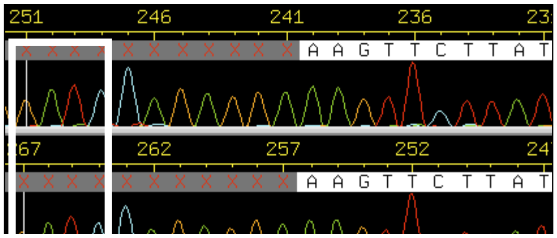


Figure 2: My misidentified clone ends. The white box outlines the true start of the clone.

My next goal was to close the gaps to form a continuous contig, which proved to be the biggest challenge of my project. Initially, I attempted to find sequence matches on the ends of neighboring contigs to force a join. I was unsuccessful, and therefore, reads were called to span the gaps. Using Consed's primer picking software, I ordered eight oligos, one read from each side of the gaps between neighboring contigs (Table 1). Due to time constraints, I needed these reads immediately, so I ordered these oligos to be used with all three possible chemistries (Big Dye, dGTP, 4:1) and on multiple templates (if possible) in case some reactions failed.

This first round of reads was compared with the reads called by Autofinish, the automated finishing program. Autofinish and I made similar calls, though Autofinish called two fewer reads (Table 2). Both my and Autofinish's oligos are designed to generate reads that will span the gaps in the construct—Autofinish is able to call two fewer reads because, unlike me, it does not call for reads from both sides of the gap. In an ideal system where all reads are of good quality and extend 600 to 700 base pairs, Autofinish's calls might be sufficient to resolve the gaps. However, sequencing is not always optimal, so it is necessary to call reads from both ends to get enough data to cover the gap between contigs. For example, one gap is too large (~900 bps) to be spanned by just one read as Autofinish attempts to do. For projects where there is sufficient time to order multiple rounds of reads, using Autofinish will help save on sequencing costs. After all, Autofinish is conservative in how many reads it calls. Plus, Autofinish is more accurate—I mistakenly called oligo 2 off an incorrect subclone, whereas Autofinish specifies the correct template. However, in our case, where time was the major constraint, Autofinish is likely too conservative.

A second problem is that Assembly View predicted that my fosmid is only 36 Kb long, including gaps. Because our fosmids should contain 38-40 Kb of sequence, I checked whether the contig had repeat structures that had been mis-assembled, resulting in a shorter contig. First, I viewed the organization of the repeats in the contig by using Consed Crossmatch. By doing so, I was able to find those regions of the contig that contained repetitive DNA sequence (Fig. 3). I then scanned those regions of the contig to search for high-quality discrepancies, as they often are indicative of misassembled repeats. As I was unable to find any, I concluded that the contig is just abnormally short and not misassembled.

While waiting for new reads, I surveyed the assembly for low consensus quality sequence, high quality discrepancies, and high quality unaligned sequence. Consed allows one to identify these regions through the navigation windows, making this task straightforward.

oligo number	oligo sequence	directionality	purpose	success	chemistry		templates	
Reads 1								
4	tcgggaaatattgtaatggac	reverse	gap (contig 16 and 8)	no	all 3	aaf02a12	aaf03b04	
5	ctcgcaactgacagcagta	forward	gap (contig 8 and 15)	no	all 3	aaf05d01	aaf05e04	
8	aatcaagggatctcattagacc	reverse	gap (contig 15 and 10)	no	all 3	aaf04e10		
9	tggaatggaagtcatataaacttg	forward	gap (contig 12 and 16)	yes	all 3	aaf02c05	aaf03f12	
7	cctgaaaatgaatgtaaggga	forward	gap (contig 15 and 10)	no	all 3	aaf04e10	aaf05e09	aaf03d10
6	gcactaggaggacatacatctaaaa	reverse	gap (contig 15 and 8)	yes	all 3	aaf05e04	aaf05d01	aaf02c04
2	tgcgaaggcactaggat	reverse	gap (contig 12 and 16)	no	all 3	aaf03f12		
3	tgctctcttaagtaatcgtaatcg	forward	gap (contig 16 and 8)	yes	all 3	aaf02a12		
Reads 2								
10	tcaccgaattttacctaattca	forward	areas of low coverage	no	4:1	aaf03b04		
11	gacaatatttccatctgccat	reverse	areas of low coverage	yes	4:1	aaf05e04		
12	accaaacttgacatatagcttctaa	reverse	areas of low coverage	yes	4:1	aaf05e04		
13	tgttaagtagagtcgacagcaagta	forward	areas of low coverage	yes	4:1	aaf05e04		
14	cgctgaaacacgtattcattatatt	reverse	areas of low coverage	yes	4:1	aaf04e10		
15	gcattctcctttctggaaaa	reverse	areas of low coverage	no	4:1	aaf04e10		
16	aacttggggctctgttaaat	forward	areas of low coverage	yes	4:1	aaf03g09		
17	aatcgagcagttcgaatctt	reverse	areas of low coverage	yes	4:1	aaf02a03		
18	tgaaattttagatgtatgtcctcct	forward	areas of low coverage	yes	4:1	aaf05d01		
19	gatacacgtaatacataattgtcca	reverse	areas of low coverage	yes	4:1	aaf02c01		
2	tgcgaaggcactaggat	reverse	gap (contig 12 and 16)	yes	4:1	aaf02c05		
20	gcagcacttgtcttatttacataat	reverse	areas of low coverage	yes	4:1	aaf02c01		
21	ttaaaggaaacgaaagacactta	reverse	areas of low coverage	yes	4:1	aaf02c01		
22	ggagcgacgactaatggata	reverse	areas of low coverage	yes	4:1	aaf05f03		
23	aattgatcaacttaaactgcataa	forward	areas of low coverage	yes	4:1	aaf05f03		
24	tctgttcctgttaaagttaattgat	reverse	areas of low coverage	yes	4:1	aaf03f12		
25	tggttttaatcaacgataactctat	reverse	areas of low coverage	yes	4:1	aaf03f12		
26	ggagcgacctacttcacca	reverse	areas of low coverage	yes	4:1	aaf05c04		
27	cgtttagattacgaaccaatgc	forward	gap (contig 12 and 93)	yes	4:1	aaf02c05		
28	cattaataataggacatttgcgat	forward	gap (contig 12 and 93)	yes	4:1	aaf02c05		
29	gagatccttgcgttcatacat	reverse	gap (contig 12 and 93)	yes	4:1	aaf02c05		
3	tgctctcttaagtaatcgtaatcg	forward	gap (contig 16 and 8)	yes	4:1	aaf02a12		
30	agatgagtaacggccataca	forward	gap (contig 93 and 95)	yes	4:1	aaf05d01	aaf02c04	aaf05e04
31	cgatttgaataaaatgggtaataat	forward	gap (contig 92 and 95)	yes	4:1	aaf05d01	aaf02c04	aaf05e04
32	cacaaagaaaatgcatttcaata	reverse	gap (contig 93 and 97)	yes	4:1	aaf05d01	aaf02c04	aaf05e04
5	ctcgcaactgacagcagta	forward	gap (contig 8 and 15)	yes	4:1	aaf05d01	aaf02c04	aaf05e04
Reads 3								
33	cacactgtacaggttattccca	reverse	areas of low coverage	yes	4:1	aaf05e04		
34	cgcagcgttcaaatatcct	forward	areas of low coverage	yes	4:1	aaf02c04		

Table 1: Reads ordered.

recommended primers					my primers			
primer	template	template	direction	purpose	primer	template	template	template
gggagacactacagtccacaat	aaf04e10		reverse	span 10 and 15	primer 8	aaf04e10		
cagaattgacattatcattgaaaa	aaf02c05	aaf03f12	forward	to span 12 and 16	primer 9	aaf02c05	aaf03f12	
taggaggacatacatctaaaatttc	aaf05e04	aaf05d01	reverse	to span 15 and 8	primer 6	aaf05e04	aaf05d01	aaf02c04
agctatacaaactgggctaatct	aaf04e10	aaf05e09	forward	to span 15 and 10	primer 7	aaf04e10	aaf05e09	aaf03d10
catccaaaatttccagtctcta	aaf04e10	aaf05e09	reverse	to span 12 and 16	primer 2	aaf03f12		
aattggaatacttctgctctctta	aaf02a12	aaf03b04	forward	to span 16 and 8	primer 3	aaf02a12		

Table 2: First round of reads that I ordered compared to Autofinish recommendations.

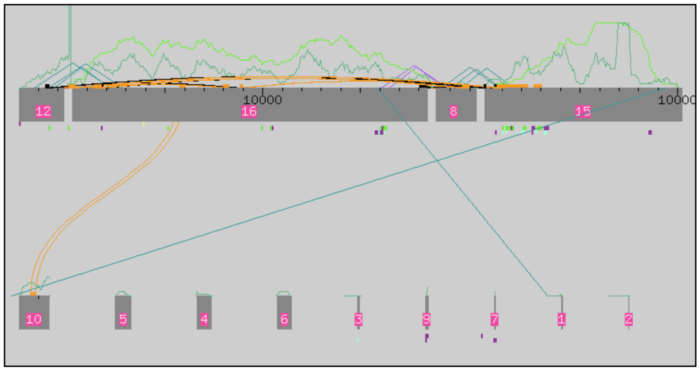
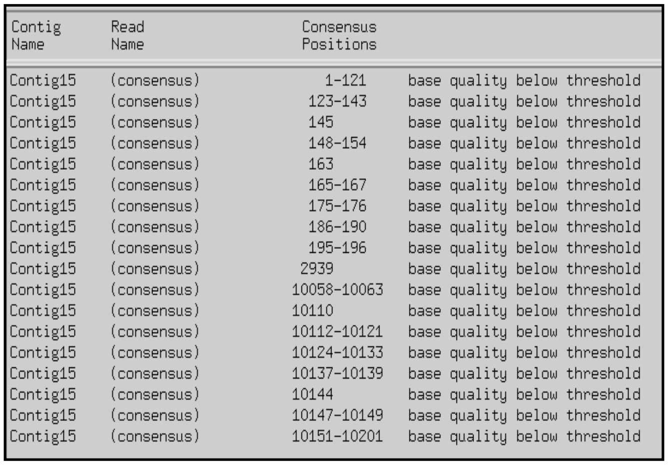
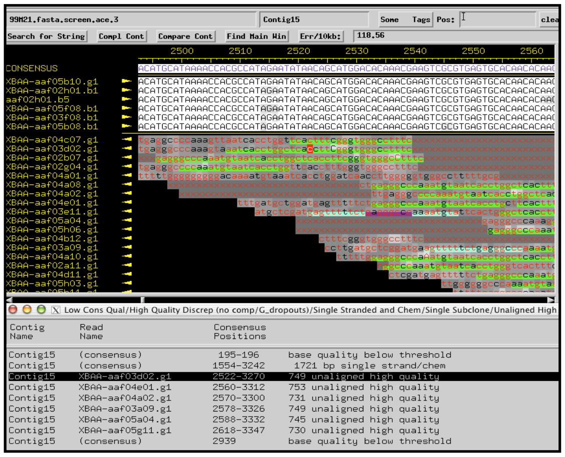


Figure 3: Contig after running Crossmatch. Orange lines represent repetitive units.



Contig Name	Read Name	Consensus Positions	
Contig15	(consensus)	1-121	base quality below threshold
Contig15	(consensus)	123-143	base quality below threshold
Contig15	(consensus)	145	base quality below threshold
Contig15	(consensus)	148-154	base quality below threshold
Contig15	(consensus)	163	base quality below threshold
Contig15	(consensus)	165-167	base quality below threshold
Contig15	(consensus)	175-176	base quality below threshold
Contig15	(consensus)	186-190	base quality below threshold
Contig15	(consensus)	195-196	base quality below threshold
Contig15	(consensus)	2939	base quality below threshold
Contig15	(consensus)	10058-10063	base quality below threshold
Contig15	(consensus)	10110	base quality below threshold
Contig15	(consensus)	10112-10121	base quality below threshold
Contig15	(consensus)	10124-10133	base quality below threshold
Contig15	(consensus)	10137-10139	base quality below threshold
Contig15	(consensus)	10144	base quality below threshold
Contig15	(consensus)	10147-10149	base quality below threshold
Contig15	(consensus)	10151-10201	base quality below threshold

Figure 4: Example of low consensus quality regions. All regions but one are located at the ends of the contig.



Contig Name	Read Name	Consensus Positions	
Contig15	(consensus)	195-196	base quality below threshold
Contig15	(consensus)	1554-3242	1721 bp single strand/chem
Contig15	XBAA-aaf03d02.g1	2522-3270	749 unaligned high quality
Contig15	XBAA-aaf04e01.g1	2560-3312	753 unaligned high quality
Contig15	XBAA-aaf04a02.g1	2570-3300	731 unaligned high quality
Contig15	XBAA-aaf03a09.g1	2578-3326	749 unaligned high quality
Contig15	XBAA-aaf05a04.g1	2588-3332	745 unaligned high quality
Contig15	XBAA-aaf05g11.g1	2618-3347	730 unaligned high quality
Contig15	(consensus)	2939	base quality below threshold

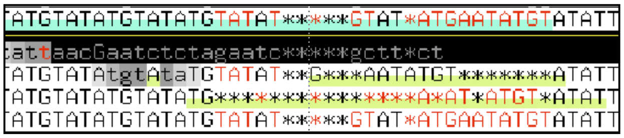
Figure 5: Example of inserted vector sequence that led to high quality, unaligned regions.

Most low quality sequence is near the ends of the individual contigs where read coverage is low. Because much of this sequence is of poor quality, the only way to resolve these areas is to order new reads across the region (examples in Fig. 4). Conveniently, all of the reads I designed to span the gaps would also help me improve consensus sequence quality in these regions.

Regions of high quality discrepancy or high quality unaligned sequences are often due to (1) vector sequence that is not recognized and (2) low sequence fidelity across a microsatellite region. As discussed earlier, the sequence of the new vector used for this project had not been entered into the GSC computers, so Phred failed to recognize vector sequence and inserted it into the contig incorrectly. This results in many discrepant areas (as shown by the sequences highlighted green in Fig. 5), which I handled by two main approaches. One method was to identify on each read where vector sequence began and to change it to Xs. I would then tear the contig at the position where the sequence became vector sequence, resulting in two separate contigs. Next, I would reassemble the same two contigs via a force join. With the vector sequence labeled appropriately, these misaligned regions pull apart and assemble correctly. Another method was to pull the discrepant reads out manually, putting them into their own contigs. Doing so resolves most "high quality discrepancy" and" high quality unaligned sequence" in the contigs.

Another major issue was that reads covering the microsatellite region do not all copy the same number of microsatellite repeats. Because the reads are then of different lengths, this leads to a region of many high-quality discrepancies (Fig. 6). To resolve this problem, I focused on the two reads that flanked the high-repeat region. Reads that begin within the repeat region are less likely to be reliable measures of microsatellite length. The two flanking trace reads were checked, and they seem to be both accurate and dependable (Fig. 7). These reads are congruent with each other and are also the longest, making it likely that this sequence is correct. To finish this area, I tore the contig on either side of the discrepant area and then used Mini-Assembly to assemble the reads again. Using Mini-Assembly allowed me to easily change the consensus sequence in this problem region.

After completing this initial survey of the problem regions of the contigs, I was able to add the first round of reads to the assembly. I had removed many reads, and using phred and phrap to add the reads would result in these changes being lost. So, I added the reads as a .fof file. A .fof file (or 'file of files') allows one to add specific reads to a project, without requiring the project to be re-assembled. Unfortunately, many of the reads seem to be of poor quality, as many of them cannot be added to the assembly.   This might be due to bad template, poorly designed primers, or inferior reaction conditions (Fig. 8).   Despite this poor overall quality, I was able to make two joins and to obtain additional data to close the two unresolved gaps.   To join the contigs, I searched for unique sequence on complementary ends of two contigs.   Doing so allowed me to make two high-quality joins between contigs 10 and 15 and contigs 8 and 16 (Fig. 9a and b). Following this round of reads, the assembly looks as seen in Figure 10.



```
TATGTATATGTATATGTATAT*****GTAT*ATGAATATGTATATT
tattaacGaatctctagaatc*****gctt*ct
TATGTATAtgtAtaTGTATAT**G***AATATGT*******ATATT
TATGTATATGTATATG**************A*AT*ATGT*ATATT
TATGTATATGTATATGTATAT*****GTAT*ATGAATATGTATATT
```

Figure 6: Discrepant microsatellite region.

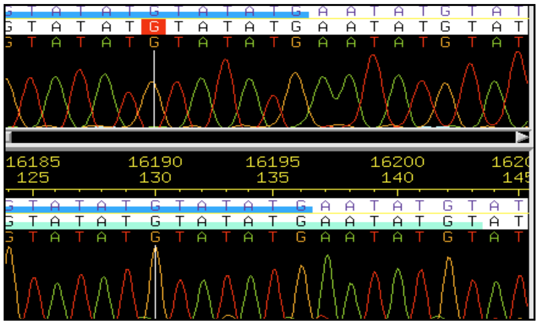


Figure 7: Dependable and congruent reads of the microsatellite region.

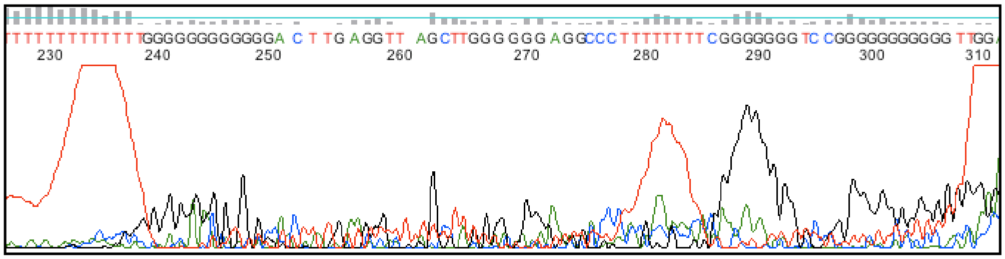
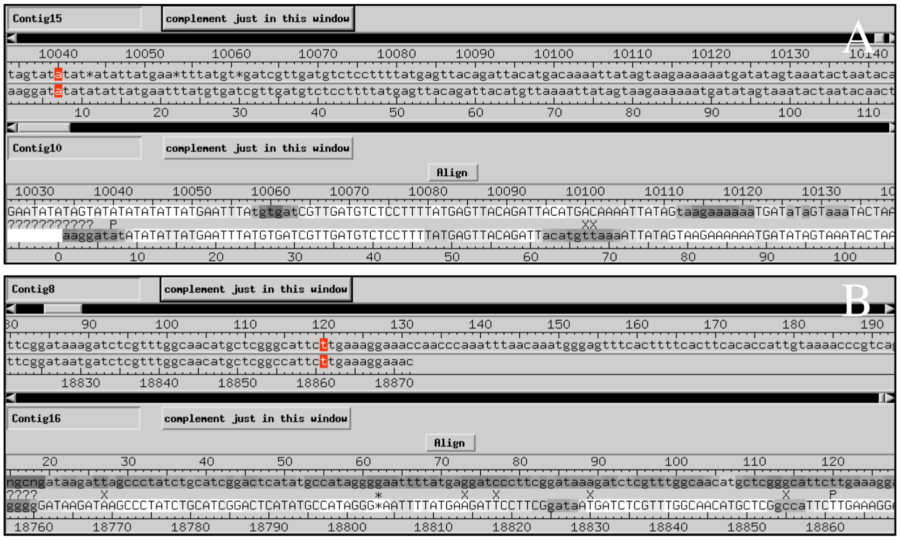


Figure 8: A good representation of the data quality of most of the generated reads.


Contig15
complement just in this window
A
Contig10
complement just in this window
Align
Contig8
complement just in this window
B
Contig16
complement just in this window
Align

```
   10040     10050     10060     10070     10080     10090     10100     10110     10120     10130     10140
tagtatatat*atattatgaa*tttatgt*gatcgttgatgtctccttttatgagttacagattacatgacaaaattatagtaagaaaaaatgatatagtaaatactaataca
aaggatatatatattatgaatttatgtgatcgttgatgtctcctttttatgagttacagattacatgttaaaattatagtaagaaaaaatgatatagtaaatactaatacaact
      10        20        30        40        50        60        70        80        90        100       110

 10030     10040     10050     10060     10070     10080     10090     10100     10110     10120     10130     10
GAATATATAGTATATATATATTATGAATTTAtgtgatCGTTGATGTCTCCTTTTATGAGTTACAGATTACATGACAAAATTATAGtaagaaaaaaTGATaTaGTaaaTACTAA
???????????   P                                                                XX
       aaggatatATATATTATGAATTTATGTGATCGTTGATGTCTCCTTTTATGAGTTACAGATTacatgttaaaATTATAGTAAGAAAAAATGATATAGTAAATACTAA
      0         10        20        30        40        50        60        70        80        90        100
```

```
30        90        100       110       120       130       140       150       160       170       180       190
ttcggataaagatctcgtttggcaacatgctcgggcattcttgaaaggaaaccaacccaaatttaacaaatgggagtttcacttttcacttcacaccattgtaaaacccgtcag
ttcggataatgatctcgtttggcaacatgctcggccattcttgaaaggaaac
         18830     18840     18850     18860     18870

    20        30        40        50        60        70        80        90        100       110       120
ngcngataagattagccctatctgcatcggactcatatgccataggggaatttatgaggatcccttcggataaagatctcgtttggcaacatgctcgggcattcttgaaagga
????        X                          *         X   X          X                        X   P
ggggGATAAGATAAGCCCTATCTGCATCGGACTCATATGCCATAGGG*AATTTTATGAAGATTCCTTCGgataATGATCTCGTTTGGCAACATGCTCGgccaTTCTTGAAAGGA
 18760     18770     18780     18790     18800     18810     18820     18830     18840     18850     18860
```

Figure 9: (A) Force join between contigs 10 and 15. (B) Force join between contigs 8 and 16.

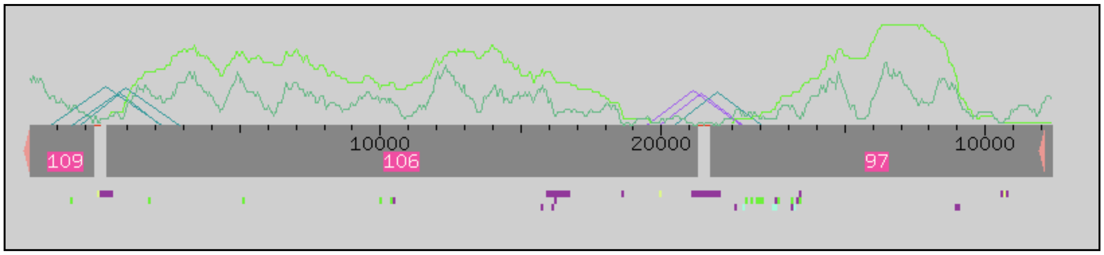
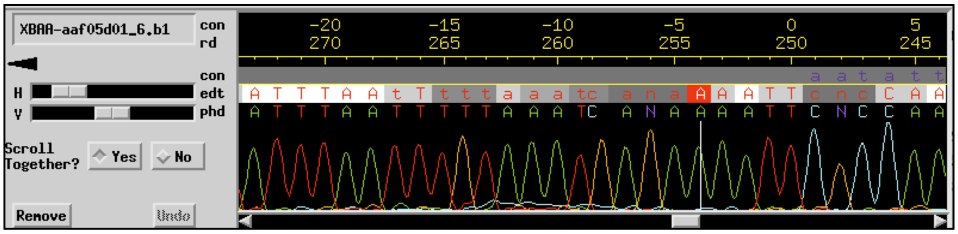


Figure 10: Assembly view after the addition of the first set of reads.

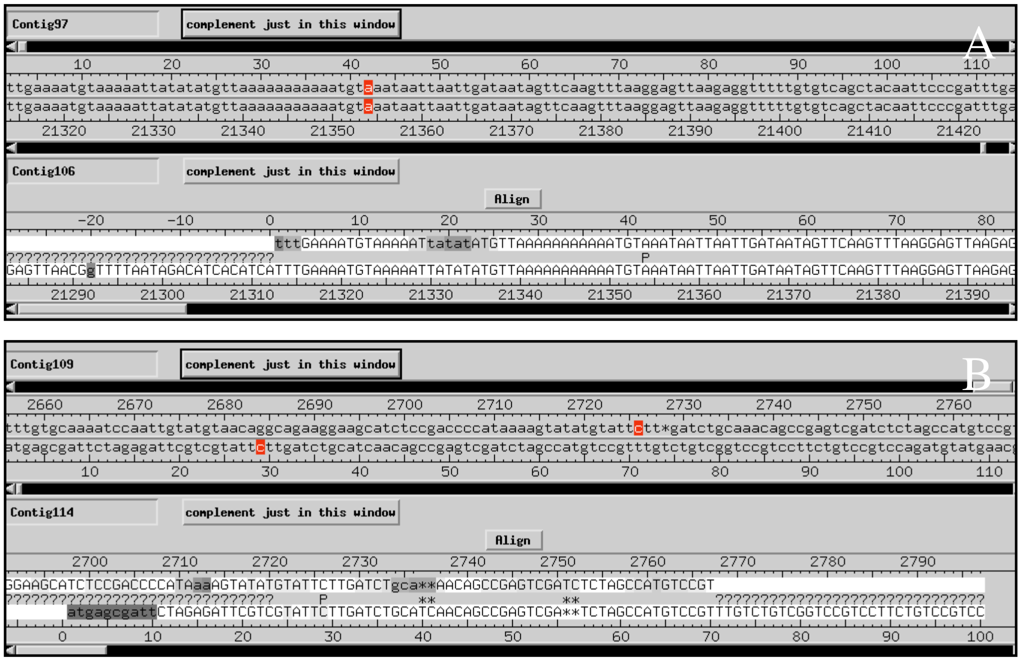
The called reads are too short to span the other two gaps, so I used primer-walking to close these gaps. In primer-walking, one successively designs primers from generated reads to extend out the sequence. However, the reads that I wanted to use to design the primers were base-called incorrectly by Phred due to large dye blobs at the start of the sequence. Because designing a well-annealing primer requires high-fidelity sequence, I pulled out these reads manually, edited them based on their traces, and then reassembled them back into the main contig (example in Fig. 11). Doing so was time-consuming, but it extends the consensus sequence into the gap, increasing the likelihood that the next set of reads will close the gap. Another problem I faced when designing primers is that some of the sequence data at the end of one of the contigs is vector, even though it is not tagged as such. Luckily, a finisher identified this region as vector based on some key sequence patterns, preventing me from designing a primer based on this extraneous sequence data. Despite these difficulties, I was able to design primers to span the two remaining gaps (Table 1). Again, due to the time constraints of our project, I designed several primers for each contig and ran them on several templates in case some reactions, primers, or templates were sub-par. In this second batch of reads, I also ordered reads to help cover regions of low sequence quality and single-stranded regions.



```
XBAA-aaf05d01_6.b1  con   -20   -15   -10   -5   0   5
                    rd    270   265   260   255  250 245
                    con                       a a t a t t
                    edt   A T T T A A t T t t t a a a tc a n a A A A T T c n c C A A
                    phd   A T T T A A T T T T T A A A TC A N A A A A T T C N C C A A
```

Figure 11: Example of discrepant base calling caused by dye blob upstream of this sequence.

Unlike the first set of reads, the majority of the second-round reads were successful. Because I used many of the same templates and some of the same primers that failed in the first set of reads, these reactions are likely successful because reaction conditions were better. These reads allowed me to close the remaining gaps and improve the coverage of the consensus. To close the gaps, I first had to set the newly generated reads as the consensus sequence by telling Consed on what trace it should base the consensus sequence. Then, I was able to make force joins between contigs 106 and 97 and contigs 109 and 112 (Figure 12a and b). Finally, the project is in one contig. The remaining reads in the second round worked to varying levels, but most provide additional coverage of low consensus regions and single-strand sequence regions. Even after adding these reads, two low-quality consensus regions still remain—one region of 500 bps beginning at 21 KB and one region of 600 bps beginning at 25 KB. Due to a misunderstanding, I was only able to order reads for the region at 25 KB in the third round of ordering reads (Table 1).



```
ttgaaaatgtaaaaattatatatgttaaaaaaaaaaaatgtaaataattaattgataatagttcaagtttaaggagttaagaggtttttgtgtcagctacaattcccgatttga
ttgaaaatgtaaaaattatatatgttaaaaaaaaaaaatgtaaataattaattgataatagttcaagtttaaggagttaagaggtttttgtgtcagctacaattcccgatttga

                             tttGAAAATGTAAAAATtatatATGTTAAAAAAAAAAAATGTAAATAATTAATTGATAATAGTTCAAGTTTAAGGAGTTAAGAG
??????????????????????????????                                         P
GAGTTAACGgTTTTAATAGACATCACATCATTTGAAAATGTAAAAATTATATATGTTAAAAAAAAAAAATGTAAATAATTAATTGATAATAGTTCAAGTTTAAGGAGTTAAGAG

tttgtgcaaaatccaattgtatgtaacaggcagaaggaagcatctccgaccccataaaagtatatgtattctt*gatctgcaaacagccgagtcgatctctagccatgtccg
atgagcgattctagagattcgtcgtattcttgatctgcatcaacagccgagtcgatctagccatgtccgtttgtctgtcggtccgtccttctgtccgtccagatgtatgaacg

GGAAGCATCTCCGACCCCATAaaAGTATATGTATTCTTGATCTgca**AACAGCCGAGTCGATCTCTAGCCATGTCCGT
??????????????????????????????        P          **           **         ??????????????????????????????
      atgagcgattCTAGAGATTCGTCGTATTCTTGATCTGCATCAACAGCCGAGTCGA**TCTAGCCATGTCCGTTTGTCTGTCGGTCCGTCCTTCTGTCCGTCC
```

Figure 12: (A) Force join between contigs 97 and 106. (B) Force join between contigs 109 and 114.

In the second round of surveying regions for "low consensus quality sequence", "high quality discrepancies", and "unaligned high quality sequences", I came across three main problems. First, Consed often determines consensus sequence using a low quality read even when high quality reads are available. This occurs especially when new reads are added as a .fof file, as discussed earlier. To remedy this, I viewed the traces of the high quality reads and made them the consensus if they seemed reliable and valid (Fig. 13). Second, there are many high quality discrepancies due to the new reads disagreeing with the old reads in determining the consensus sequence (Fig. 14). These discrepancies are resolved easily by editing the bases to reflect the most consistent high quality base call. Third, many regions of high quality unaligned sequences are mislabeled. Some regions are labeled high quality, but, in reality, the base reads are undependable due to multiple peaks (Fig. 15). These regions were tagged and left alone. Consed identified other regions as high quality unaligned sequences even though their sequence traces were highly congruent with the consensus sequence. These reads were removed from the contig and then reinserted, after which Consed no longer identified them as high quality unaligned.

To check the quality of the consensus sequence, I ran two *in silico* digests to compare to the original restriction digests done on my fosmid. These digests illuminate another remaining problem. As can be seen in the *Eco*RV digest, there are three anomalous bands (Fig. 16a). The two unmatched bands at 2476 and 2527 are not truly discrepant—upon a visual examination of the actual digest, the corresponding bands are actually doublets. However, the discrepancy in the band at 2965 base pairs indicates a problem with the consensus. I strongly suspect that this band should correspond with the actual fragment of 3211 bps, because it is the closest fragment that does not have a match in the *in silico* digest. This represents an approximately 240 bp discrepancy in the 25.8-28.8 Kb region. Viewing a second digest done with *Hind*III again shows three discrepant bands. The band-calling software misread the fragment of 1908 bps seen

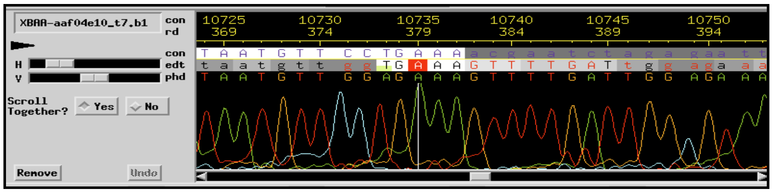
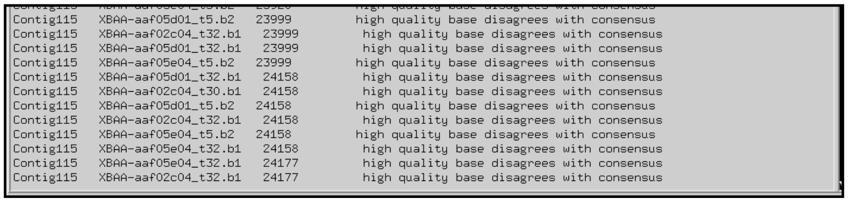

XBAA-aaf04e10_t7.b1
con
rd
H
V
con
edt
phd
Scroll Together?
Yes
No
Remove
Undo
10725 369
10730 374
10735 379
10740 384
10745 389
10750 394
T A A T G T T C C T G A A A a c g a a t c t a g a g a a t t
t a a t g t t g g T G A A A G T T T T G A T t g g a g a a a
T A A T G T T G G A G A A A G T T T T G A T T G G A G A A A

Figure 13: Example of consensus sequence being called off of a poor quality read when better data existed.



```
Contig115   XBAA-aaf05d01_t5.b2   23999         high quality base disagrees with consensus
Contig115   XBAA-aaf02c04_t32.b1   23999         high quality base disagrees with consensus
Contig115   XBAA-aaf05d01_t32.b1   23999         high quality base disagrees with consensus
Contig115   XBAA-aaf05e04_t5.b2   23999         high quality base disagrees with consensus
Contig115   XBAA-aaf05d01_t32.b1   24158         high quality base disagrees with consensus
Contig115   XBAA-aaf02c04_t30.b1   24158         high quality base disagrees with consensus
Contig115   XBAA-aaf05d01_t5.b2   24158         high quality base disagrees with consensus
Contig115   XBAA-aaf02c04_t32.b1   24158         high quality base disagrees with consensus
Contig115   XBAA-aaf05e04_t5.b2   24158         high quality base disagrees with consensus
Contig115   XBAA-aaf05e04_t32.b1   24158         high quality base disagrees with consensus
Contig115   XBAA-aaf05e04_t32.b1   24177         high quality base disagrees with consensus
Contig115   XBAA-aaf02c04_t32.b1   24177         high quality base disagrees with consensus
```

Figure 14: Examples of how most high quality base discrepancies were caused by one bp discrepancies.

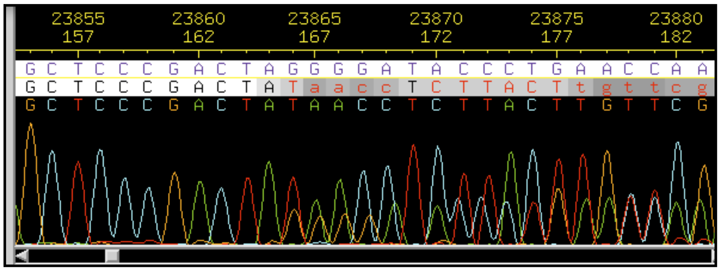
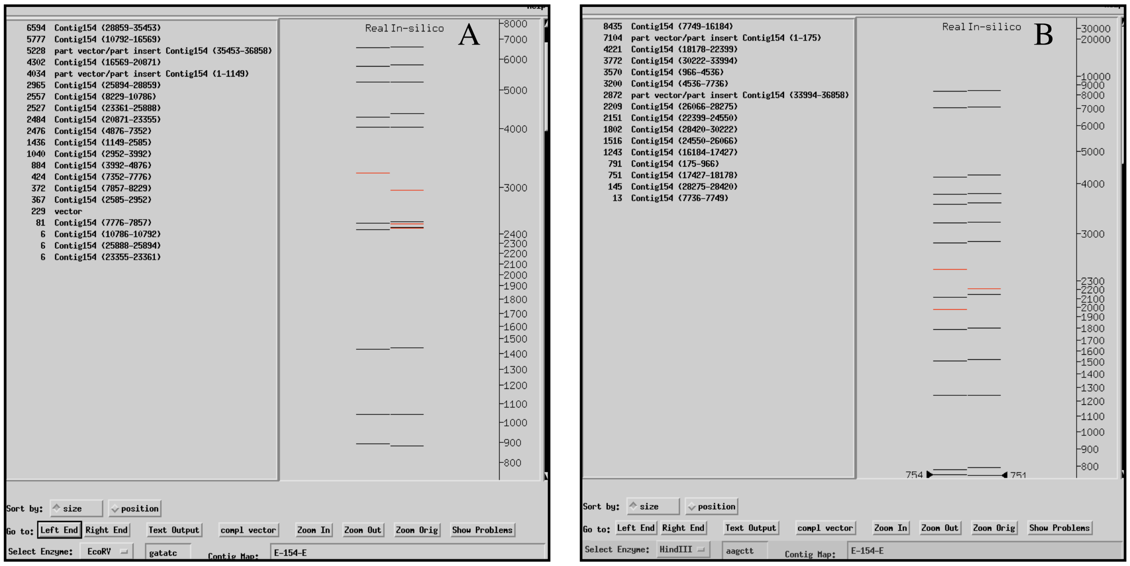

23855 157
23860 162
23865 167
23870 172
23875 177
23880 182
G C T C C C G A C T A G G G G A T A C C C T G A A C C A A
G C T C C C G A C T A T a a c c T C T T A C T t g t t c g
G C T C C C G A C T A T A A C C T C T T A C T T G T T C G

Figure 15: Case where high quality unaligned region was not that high quality.

on the real digest, and so this discrepancy is disregarded. The other discrepancy is between the real band of 2460 bps and the *in silico* band at 2209 bps. I again suspect this is because the consensus sequence is missing 240 bps in the 26-28.2 Kb region (Fig. 16B). I hypothesize both these discrepancies are due to a 240 bp missing chunk of data in the final consensus, probably due to a misassembled repeat. Indeed, this region of the contig is highly repetitious (Fig. 17).



```
6594  Contig154 (28859-35453)
5777  Contig154 (10792-16569)
5228  part vector/part insert Contig154 (35453-36858)
4302  Contig154 (16569-20871)
4034  part vector/part insert Contig154 (1-1149)
2965  Contig154 (25894-28859)
2557  Contig154 (8229-10786)
2527  Contig154 (23361-25888)
2484  Contig154 (20871-23355)
2476  Contig154 (4876-7352)
1436  Contig154 (1149-2585)
1040  Contig154 (2952-3992)
 884  Contig154 (3992-4876)
 424  Contig154 (7352-7776)
 372  Contig154 (7857-8229)
 367  Contig154 (2585-2952)
 229  vector
  81  Contig154 (7776-7857)
   6  Contig154 (10786-10792)
   6  Contig154 (25888-25894)
   6  Contig154 (23355-23361)
```

```
8435  Contig154 (7749-16184)
7104  part vector/part insert Contig154 (1-175)
4221  Contig154 (18178-22399)
3772  Contig154 (30222-33994)
3570  Contig154 (966-4536)
3200  Contig154 (4536-7736)
2872  part vector/part insert Contig154 (33994-36858)
2209  Contig154 (26066-28275)
2151  Contig154 (22399-24550)
1802  Contig154 (28420-30222)
1516  Contig154 (24550-26066)
1243  Contig154 (16184-17427)
 791  Contig154 (175-966)
 751  Contig154 (17427-18178)
 145  Contig154 (28275-28420)
  13  Contig154 (7736-7749)
```

Sort by: size position
Go to: Left End Right End Text Output compl vector Zoom In Zoom Out Zoom Orig Show Problems
Select Enzyme: EcoRV gatatc Contig Map: E-154-E

Sort by: size position
Go to: Left End Right End Text Output compl vector Zoom In Zoom Out Zoom Orig Show Problems
Select Enzyme: HindIII aagctt Contig Map: E-154-E

Figure 16: (A) *Eco*RV actual and *in silico* digest. (B) *Hind*III actual and *in silico* digest.

To reassemble the contig correctly, I attempted several techniques. I first characterized the repeat structure by doing multiple *search for strings* and noting patterns seen in the searches. This helped me determine unique regions in the repeats, helping target my search for discrepancies (Fig. 18). I also determined that the repeat is of approximately 270 bps, which fits in well with the length incongruities in the restriction maps. I then tore the contig on either side of the repeat region and attempted to reassemble this contig using tighter phred standards. This was unsuccessful, so I then scanned all 2 Kb for any regions of high or low quality discrepancies. Discrepancies often mark two unique repeats that have been mistakenly assembled with each other. I was unable to find any, so I concluded that this region requires more reads to find the missing repeat.

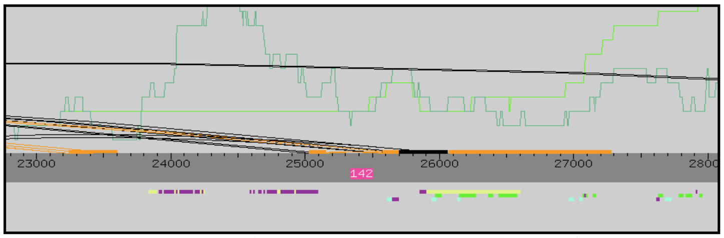


Figure 17: Highly repetitious sequence in problem region.

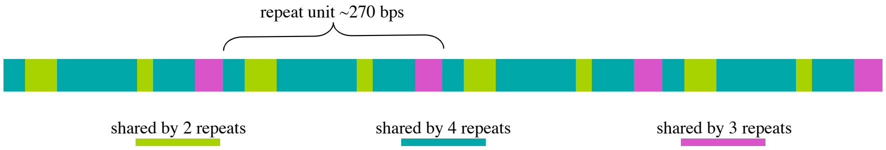


Figure 18: Structure of repeat units in the problem region.

Before I could determine the final consensus sequence, I needed to confirm that I had no remaining problems. Many of the problems that finishers often face are not applicable to my project. For example, there are many unincorporated contigs but they are small and mainly contain unidentified vector sequence. I did not need to do a BLAST search as I found no evidence of contamination, a finding that was corroborated by a problem-free Findid report. The consensus sequence is free of unidentified nucleotides (Ns), vector sequence (Xs), and mononucleotide runs. In the end, there are three remaining areas of concern: 500 bps of poor consensus quality at 21 Kb, a missing repeat unit in the 26-28.2 Kb region, and some regions still covered by single-strand reads. Single-strand coverage is acceptable by the standards (*Mus musculus* genome) to which we are finishing this genome, but the other two regions will require additional reads to get final resolution. I tagged these regions appropriately, and other finishers will complete these regions prior to this fosmid's annotation. Figure 19 shows my final contig.

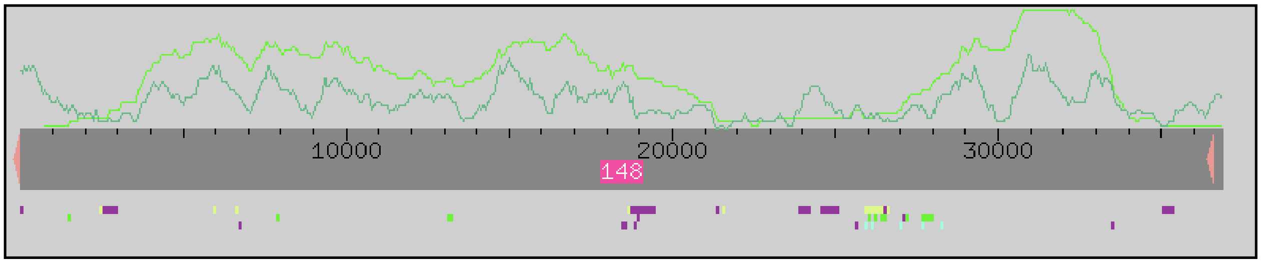


Figure 19: Final contig.