PROBE MINING WITH OLIGOARRAY

SEARCHING THE DROSOPHILA HOX CLUSTERS FOR 50MER PROBE SEQUENCES

By Brian Beliveau

Last updated 8/27/12

Contact: oligopaints@genetics.med.harvard.edu

INTRODUCTION

This guide gives an overview of how to discover FISH probe sequences using the program OligoArray and assumes the user has already installed python and OligoArray along with its dependency OligoArrayAux. OligoArray and its installation instructions can be found here, and OligoArrayAux and its installation instructions can be found here. Instructions for downloading and installing python can be found in our Oligopaints Scripts Manual. Note that this example is tailored to the LSF system employed by the Harvard RITG Orchestra UNIX cluster; users will need to modify the paths specified in this example (e.g. /files/Genetics/Wu Lab/Oligopaints/hox_example/) to match the desired working directories on their home machines or research clusters.

GETTING STARTED

The Drosophila Hox genes are broken into two ~300 kb clusters on chr3R – the antennapedia cluster which spans from *labial* to *antennapedia* and the bithorax cluster which spans from *ultrabithorax* to *abdominal B*. In dm3 coordinates, the antennapedia cluster spans from 2,487,149 - 2,824,950 and the bithorax cluster spans from 12,482,345 - 12,797,958. We can expand our target region to include flanking regions as well, such that we will mine a 1 Mb region for each target. In order to run OligoArray, we need to prepare two types of files: input files containing the sequence we want to mine for probes and a BLAST database which will be used to asses the specificity of any candidate probe sequences.

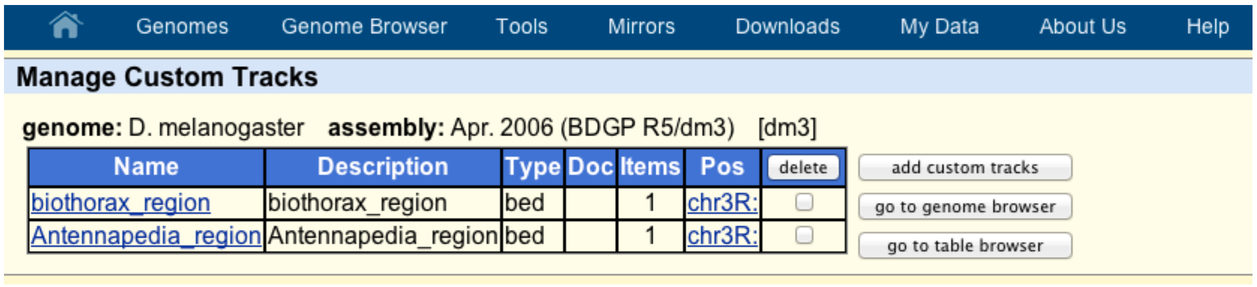
PREPARING INPUT FILES

First, we will download fasta format files for each region we want to mine from the UCSC genome browser. For each cluster, we will prepare a .bed files specifying the sequences we want to download. These files will look like this:

```
track name="Antennapedia_region"
chr3R 2156050	3156050
```

```
track name="biothorax_region"
chr3R 12140152      13140152
```

We can upload these files as Custom Tracks to UCSC:



Manage Custom Tracks

genome: D. melanogaster **assembly:** Apr. 2006 (BDGP R5/dm3) [dm3]

Name	Description	Type	Doc	Items	Pos	delete
biothorax_region	biothorax_region	bed		1	chr3R:	☐
Antennapedia_region	Antennapedia_region	bed		1	chr3R:	☐

We can then go to the table browser and use these .bed files to download sequence files:

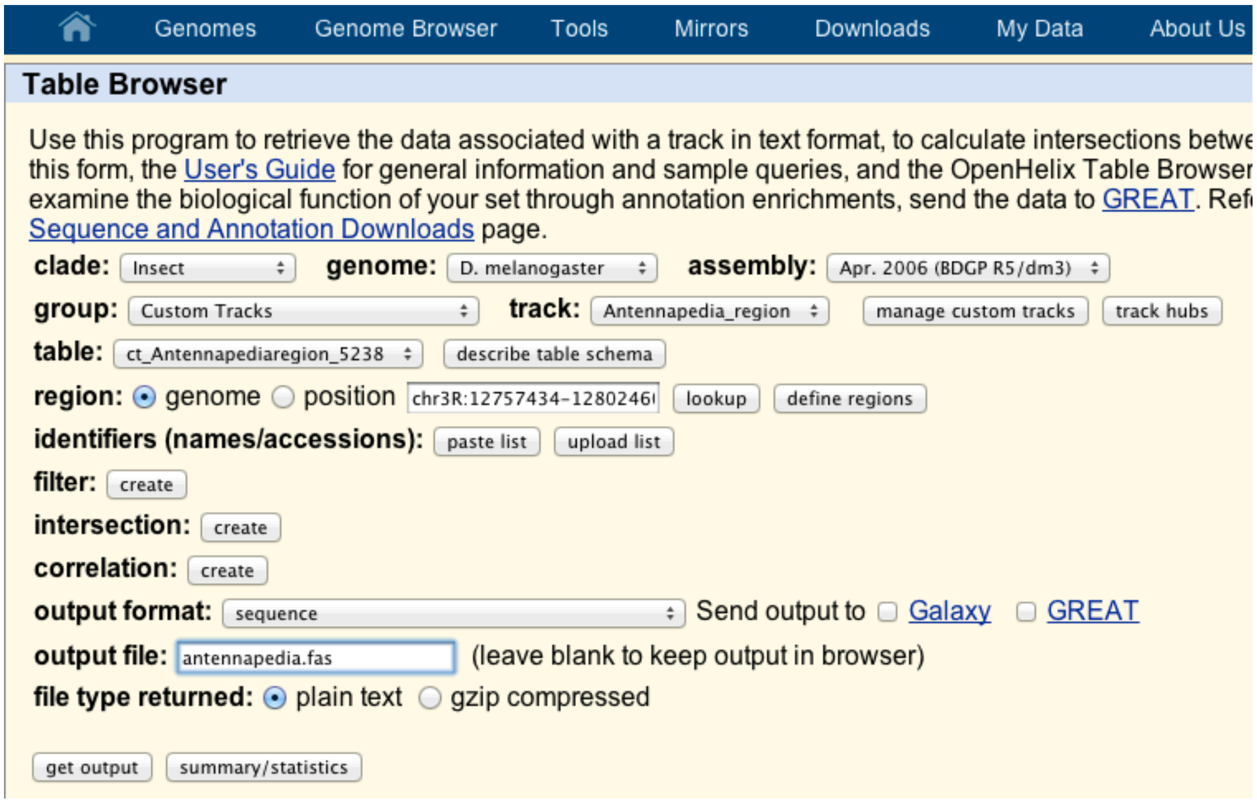


Table Browser

Use this program to retrieve the data associated with a track in text format, to calculate intersections betwe
this form, the User's Guide for general information and sample queries, and the OpenHelix Table Browser
examine the biological function of your set through annotation enrichments, send the data to GREAT. Ref
Sequence and Annotation Downloads page.

clade: Insect **genome:** D. melanogaster **assembly:** Apr. 2006 (BDGP R5/dm3)

group: Custom Tracks **track:** Antennapedia_region

table: ct_Antennapediaregion_5238

region: [✓ genome] position chr3R:12757434-1280246

identifiers (names/accessions):

filter:

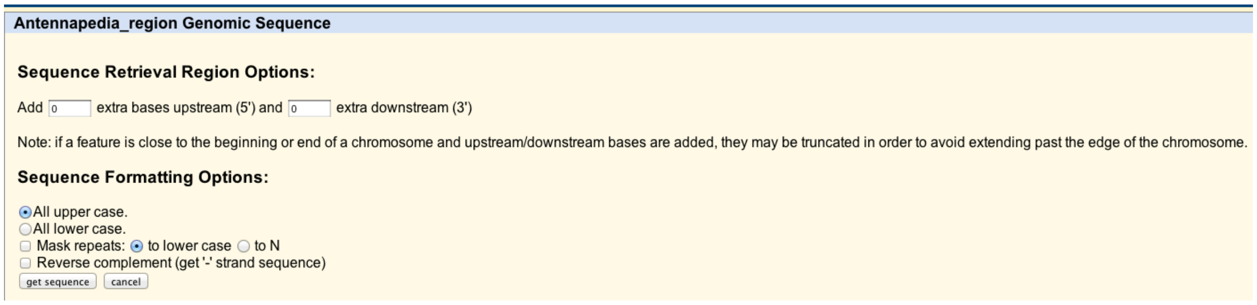
intersection:

correlation:

output format: sequence Send output to ☐ Galaxy ☐ GREAT

output file: antennapedia.fas (leave blank to keep output in browser)

file type returned: [✓ plain text] gzip compressed



Antennapedia_region Genomic Sequence

Sequence Retrieval Region Options:

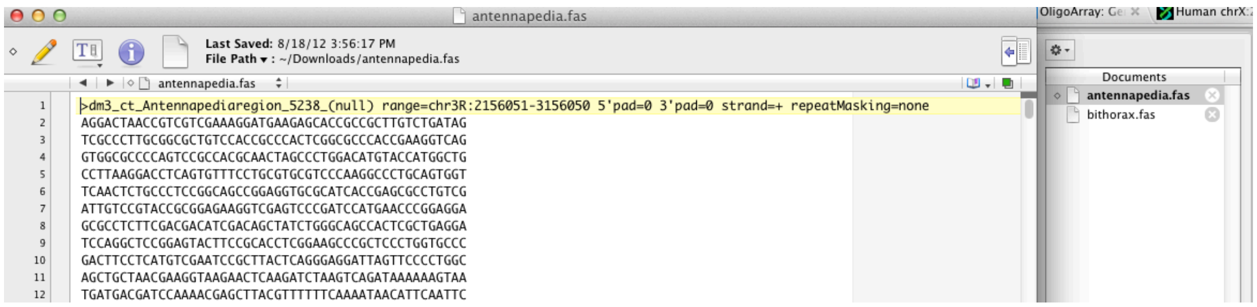
Add 0 extra bases upstream (5') and 0 extra downstream (3')

Note: if a feature is close to the beginning or end of a chromosome and upstream/downstream bases are added, they may be truncated in order to avoid extending past the edge of the chromosome.

Sequence Formatting Options:

[✓ All upper case.]
All lower case.
Mask repeats: [✓ to lower case] to N
Reverse complement (get '-' strand sequence)

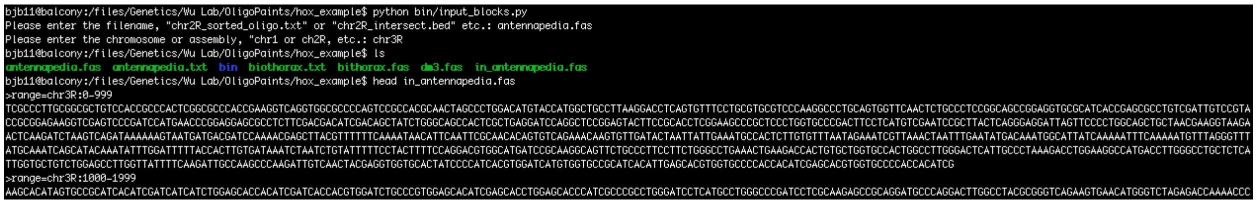
Which yields a .fas file with a fasta header and our sequence of interest:



Last Saved: 8/18/12 3:56:17 PM
File Path ▾ : ~/Downloads/antennapedia.fas

```
>dm3_ct_Antennapediaregion_5238_(null) range=chr3R:2156051-3156050 5'pad=0 3'pad=0 strand=+ repeatMasking=none
AGGACTAACCGTCGTCGAAAGGATGAAGAGCACCGCCGCTTGTCTGATAG
TCGCCCTTGCGGCGCTGTCCACCGCCCACTCGGCGCCCACCGAAGGTCAG
GTGGCGCCCCAGTCCGCCACGCAACTAGCCCTGGACATGTACCATGGCTG
CCTTAAGGACCTCAGTGTTTCCTGCGTGCGTCCCAAGGCCCTGCAGTGGT
TCAACTCTGCCCTCCGGCAGCCGGAGGTGCGCATCACCGAGCGCCTGTCG
ATTGTCCGTACCGCGGAGAAGGTCGAGTCCCGATCCATGAACCCGGAGGA
GCGCCTCTTCGACGACATCGACAGCTATCTGGGCAGCCACTCGCTGAGGA
TCCAGGCTCCGGAGTACTTCCGCACCTCGGAAGCCCGCTCCCTGGTGCCC
GACTTCCTCATGTCGAATCCGCTTACTCAGGGAGGATTAGTTCCCCTGGC
AGCTGCTAACGAAGGTAAGAACTCAAGATCTAAGTCAGATAAAAAAGTAA
TGATGACGATCCAAAACGAGCTTACGTTTTTTCAAAATAACATTCAATTC
```

In our bioinformatics pipeline, we break our input files into 1 kb chunks using the script input_blocks.py:

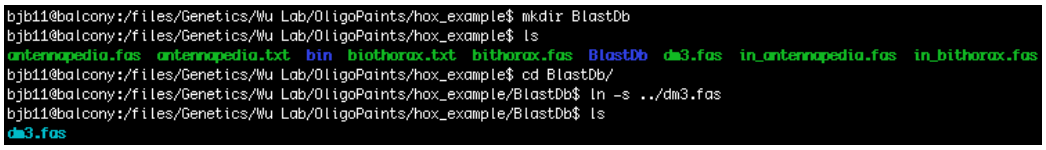


```
bjb11@balcony:/files/Genetics/Wu Lab/OligoPaints/hox_example$ python bin/input_blocks.py
Please enter the filename, "chr2R_sorted_oligo.txt" or "chr2R_intersect.bed" etc.: antennapedia.fas
Please enter the chromosome or assembly, "chr1 or ch2R, etc.: chr3R
bjb11@balcony:/files/Genetics/Wu Lab/OligoPaints/hox_example$ ls
antennapedia.fas  antennapedia.txt  bin  biothorax.txt  bithorax.fas  dm3.fas  in_antennapedia.fas
bjb11@balcony:/files/Genetics/Wu Lab/OligoPaints/hox_example$ head in_antennapedia.fas
>range=chr3R:0-999
TCGCCCTTGCGGCGCTGTCCACCGCCCACTCGGCGCCCACCGAAGGTCAGGTGGCGCCCCAGTCCGCCACGCAACTAGCCCTGGACATGTACCATGGCTGCCTTAAGGACCTCAGTGTTTCCTGCGTGCGTCCCAAGGCCCTGCAGTGGTTCAACTCTGCCCTCCGGCAGCCGGAGGTGCGCATCACCGAGCGCCTGTCGATTGTCCGTA
CCGCGGAGAAGGTCGAGTCCCGATCCATGAACCCGGAGGAGCGCCTCTTCGACGACATCGACAGCTATCTGGGCAGCCACTCGCTGAGGATCCAGGCTCCGGAGTACTTCCGCACCTCGGAAGCCCGCTCCCTGGTGCCCGACTTCCTCATGTCGAATCCGCTTACTCAGGGAGGATTAGTTCCCCTGGCAGCTGCTAACGAAGGTAAGA
ACTCAAGATCTAAGTCAGATAAAAAAGTAATGATGACGATCCAAAACGAGCTTACGTTTTTTCAAAATAACATTCAATTCGCAACACAGTGTCAGAAACAAGTGTTGATACTAATTATTGAAATGCCACTCTTGTGTTTAATAGAAATCGTTAAACTAATTTGAATATGACAAATGGCATTATCAAAAATTTCAAAAATGTTTAGGGTTT
ATGCAAATCAGCATACAAATATTTGGATTTTTACCACTTGTGATAAATCTAATCTGTATTTTTCCTACTTTTCCAGGACGTGGCATGATCCGCAAGGCAGTTCTGCCCTTCCTTCTGGGCCTGAAACTGAAGACCACTGTGCTGGTGCCACTGGCCTTGGGACTCATTGCCCTAAAGACCTGGAAGGCCATGACCTTGGGCCTGCTCTCA
TTGGTGCTGTCTGGAGCCTTGGTTATTTTCAAGATTGCCAAGCCCAAGATTGTCAACTACGAGGTGGTGCACTATCCCCATCACGTGGATCATGTGGTGCCGCATCACATTGAGCACGTGGTGCCCCACCACATCGAGCACGTGGTGCCCCACCACATCG
>range=chr3R:1000-1999
AAGCACATAGTGCCGCATCACATCGATCATCATCTGGAGCACCACATCGATCACCACGTGGATCTGCCCGTGGAGCACATCGAGCACCTGGAGCACCCATCGCCCGCCTGGGATCCTCATGCCTGGGCCCGATCCTCGCAAGAGCCGCAGGATGCCCAGGACTTGGCCTACGCGGGTCAGAAGTGAACATGGGTCTAGAGACCAAAACCC
```

This generates files ready to input into OligoArray.

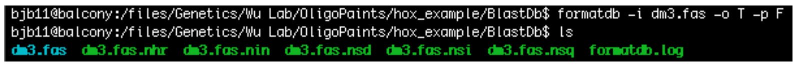
GENERATING A CUSTOM BLAST DATABASE

Now that our input files are ready, we need to make a custom BLAST database. This step only needs to be done once for each genome to be searched. The starting point for this task is a .fas file containing the entire genome of the organism in question, arranged in the same 1 kb input file format. These can be created by downloading whole-chromosome fasta files either from UCSC using a .bed or directly from a repository such as GenBank. These files can then be concatenated using the ConcatenateFiles.class script available from the OligoArray website. Once a master file for the whole genome is assembled, the input_blocks.py script can then be run. Master files for each genome already run by the Wu lab are available on the Oligopaints website. In this example, a master file for the Drosophila genome, “dm3.fas,” already exists in our working directory. We make a new folder called “BlastDb,” in which we create a symbolic link back to our dm3.fas master file:



```
bjb11@balcony:/files/Genetics/Wu Lab/OligoPaints/hox_example$ mkdir BlastDb
bjb11@balcony:/files/Genetics/Wu Lab/OligoPaints/hox_example$ ls
antennapedia.fas  antennapedia.txt  bin  biothorax.txt  bithorax.fas  BlastDb  dm3.fas  in_antennapedia.fas  in_bithorax.fas
bjb11@balcony:/files/Genetics/Wu Lab/OligoPaints/hox_example$ cd BlastDb/
bjb11@balcony:/files/Genetics/Wu Lab/OligoPaints/hox_example/BlastDb$ ln -s ../dm3.fas
bjb11@balcony:/files/Genetics/Wu Lab/OligoPaints/hox_example/BlastDb$ ls
dm3.fas
```

We can now make our BLAST database using the formatdb command with the following paramters: “-o T –p F.” Note that formatdb is dependent on an appropriate version of BLAST being installed. Also see the OligoArray website.



```
bjb11@balcony:/files/Genetics/Wu Lab/OligoPaints/hox_example/BlastDb$ formatdb -i dm3.fas -o T -p F
bjb11@balcony:/files/Genetics/Wu Lab/OligoPaints/hox_example/BlastDb$ ls
dm3.fas  dm3.fas.nhr  dm3.fas.nin  dm3.fas.nsd  dm3.fas.nsi  dm3.fas.nsq  formatdb.log
```

RUNNING OLIGOARRAY

With our input files and BLAST database in hand, we are ready to run OligoArray. See the OligoArray website for details on all the available options. In this example, OligoArray will be run on Harvard Orchestra UNIX cluster. For each region, we will create a tab-delimited .txt file specifying the parameters to run OligoArray with. The contents of an example file, “batch_antennapedia.txt,” are show below:

```
java -Xmx8192m -jar /home/bjb11/OligoArray2_1/OligoArray2.jar -i
in_atennapedia.fas -d BlastDb/dm3.fas -o antennapedia_oligo.txt -r
antennapedia_rejected.fas -R antennapedia.log -n 30 -l 50 -L 50 -D 1000 -t 85 -T
99 -s 70 -x 70 -p 35 -P 80 -m "GGGG;CCCC;TTTTT;AAAAA" -g 52
```

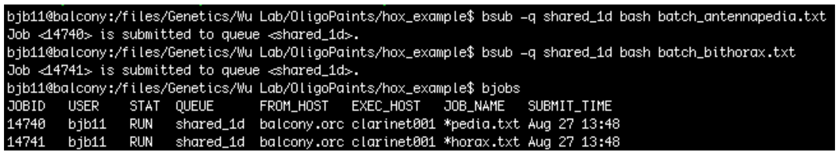
This text specifies to search each region for 50mer probe sequences with an estimated T_M of between 85 and 99°C and a GC content of 35 - 80% that do not contain the sequences “GGGG,” “CCCC,” “TTTTT,” or “AAAAA.” A minimum spacing of 2 bp between probes is enforced. OligoArray will output three files: antennapedia_oligo.txt, which will detail every probe sequence discovered; antennapedia_rejected.fas, the input sequence blocks in which 0 probes were found in a format that is ready to be re-inputted into OligoArray; antennapedia.log, which details the line-by-line activity of OligoArray.

At this point, the files can be submitted to the cluster or run on the user’s home machine. Note that is important to make sure OligoArray2.jar and our input files have the appropriate permissions. Permissions can be checked by typing “ls -l,” and changed with the chmod command.

To run OligoArray locally, the user can simply type:

```
$ bash batch_antennapedia.txt
```

With our LSF system, the command on the cluster would be:

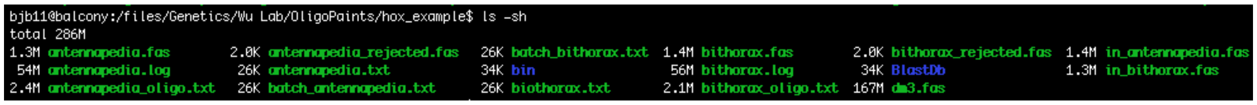


```
bjb11@balcony:/files/Genetics/Wu Lab/OligoPaints/hox_example$ bsub -q shared_1d bash batch_antennapedia.txt
Job <14740> is submitted to queue <shared_1d>.
bjb11@balcony:/files/Genetics/Wu Lab/OligoPaints/hox_example$ bsub -q shared_1d bash batch_bithorax.txt
Job <14741> is submitted to queue <shared_1d>.
bjb11@balcony:/files/Genetics/Wu Lab/OligoPaints/hox_example$ bjobs
JOBID   USER    STAT  QUEUE      FROM_HOST   EXEC_HOST   JOB_NAME   SUBMIT_TIME
14740   bjb11   RUN   shared_1d  balcony.orc clarinet001 *pedia.txt Aug 27 13:48
14741   bjb11   RUN   shared_1d  balcony.orc clarinet001 *horax.txt Aug 27 13:48
```

Be advised that OligoArray can be very computationally intensive and can generate large (gigabyte-scale) quantities of data.

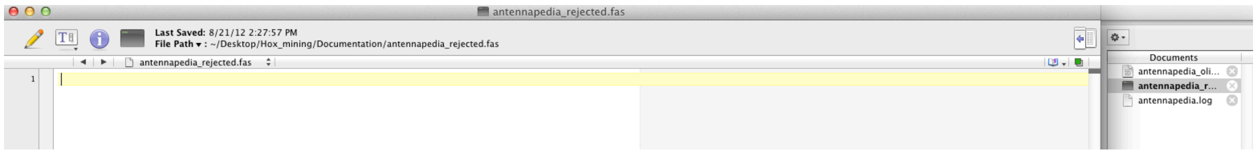
PROCESSING THE RESULTS

Once OligoArray has finished running, there were be several additional files in the working directory:



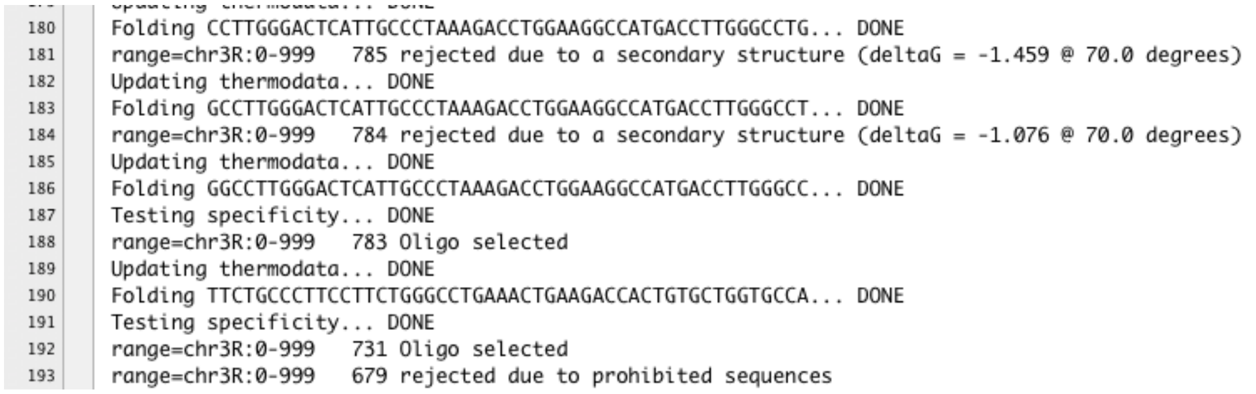
```
bjb11@balcony:/files/Genetics/Wu Lab/OligoPaints/hox_example$ ls -sh
total 286M
1.3M antennapedia.fas        2.0K antennapedia_rejected.fas  26K batch_bithorax.txt 1.4M bithorax.fas        2.0K bithorax_rejected.fas 1.4M in_antennapedia.fas
 54M antennapedia.log         26K antennapedia.txt           34K bin                 56M bithorax.log         34K BlastDb               1.3M in_bithorax.fas
2.4M antennapedia_oligo.txt   26K batch_antennapedia.txt     26K biothorax.txt      2.1M bithorax_oligo.txt  167M dm3.fas
```

Forunately, the antennepdia_rejected.fas file is empty, indicating that OligoArray was able to find at least 1 probe in every 1 kb chunk inputted:



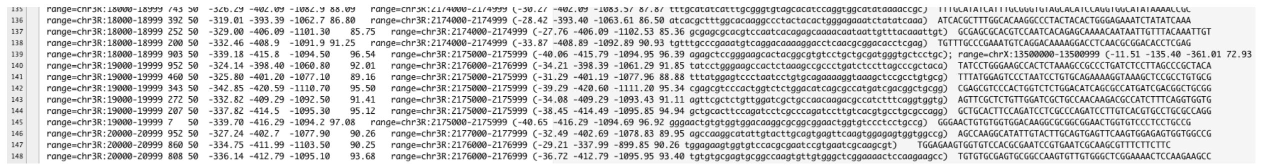
```
antennapedia_rejected.fas
Last Saved: 8/21/12 2:27:57 PM
File Path ▾ : ~/Desktop/Hox_mining/Documentation/antennapedia_rejected.fas
```

OligoArray also produced a very dense log file, a portion of which is shown below:



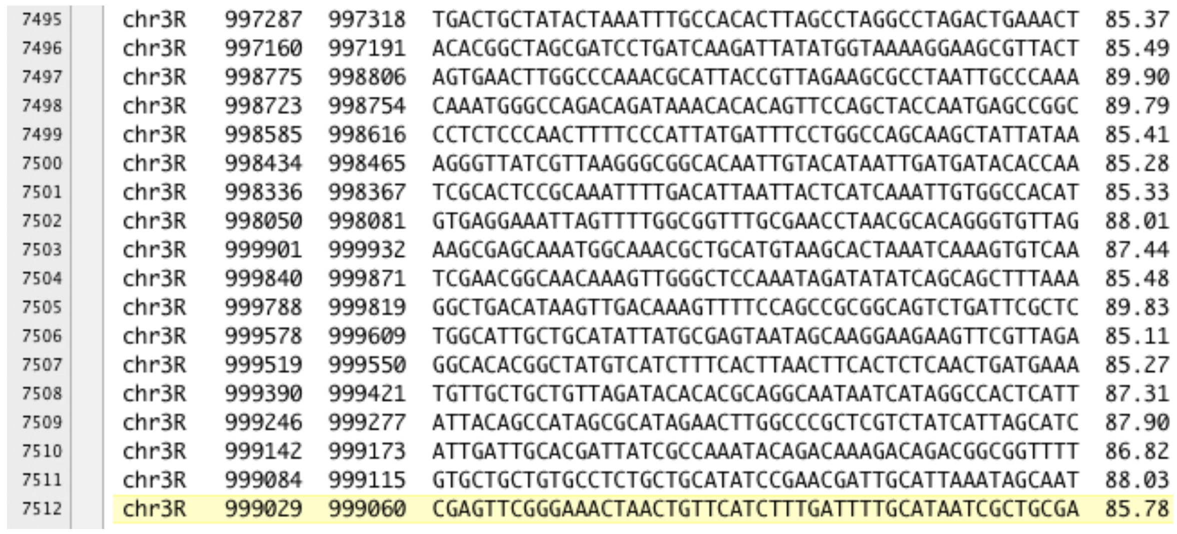
```
Folding CCTTGGGACTCATTGCCCTAAAGACCTGGAAGGCCATGACCTTGGGCCTG... DONE
range=chr3R:0-999   785 rejected due to a secondary structure (deltaG = -1.459 @ 70.0 degrees)
Updating thermodata... DONE
Folding GCCTTGGGACTCATTGCCCTAAAGACCTGGAAGGCCATGACCTTGGGCCT... DONE
range=chr3R:0-999   784 rejected due to a secondary structure (deltaG = -1.076 @ 70.0 degrees)
Updating thermodata... DONE
Folding GGCCTTGGGACTCATTGCCCTAAAGACCTGGAAGGCCATGACCTTGGGCC... DONE
Testing specificity... DONE
range=chr3R:0-999   783 Oligo selected
Updating thermodata... DONE
Folding TTCTGCCCTTCCTTCTGGGCCTGAAACTGAAGACCACTGTGCTGGTGCCA... DONE
Testing specificity... DONE
range=chr3R:0-999   731 Oligo selected
range=chr3R:0-999   679 rejected due to prohibited sequences
```

And finally, OligoArray outputs a file detaling the selected oligos and their properties:



```
135 range=chr3R:18000-18999 743 50  -326.29 -402.09 -1082.9 88.09   range=chr3R:2174000-2174999 (-30.27 -402.09 -1083.57 87.87 tttgcatatcatttgcgggtgtagcacatccaggtggcatataaaaccgc)  TTTGCATATCATTTGCGGGTGTAGCACATCCAGGTGGCATATAAAACCGC
136 range=chr3R:18000-18999 392 50  -319.01 -393.39 -1062.7 86.80   range=chr3R:2174000-2174999 (-28.42 -393.40 -1063.61 86.50 atcacgctttggcacaaggccctactacactgggagaaatctatatcaaa)  ATCACGCTTTGGCACAAGGCCCTACTACACTGGGAGAAATCTATATCAAA
137 range=chr3R:18000-18999 252 50  -329.00 -406.09 -1101.30    85.75   range=chr3R:2174000-2174999 (-27.76 -406.09 -1102.53 85.36 gcgagcgcacgtccaatcacagagcaaaacaataattgtttacaaattgt)  GCGAGCGCACGTCCAATCACAGAGCAAAACAATAATTGTTTACAAATTGT
138 range=chr3R:18000-18999 200 50  -332.46 -408.9  -1091.9 91.25   range=chr3R:2174000-2174999 (-33.87 -408.89 -1092.89 90.93 tgtttgcccgaaatgtcaggacaaaaggacctcaacgcggacacctcgag)  TGTTTGCCCGAAATGTCAGGACAAAAGGACCTCAACGCGGACACCTCGAG
139 range=chr3R:18000-18999 903 50  -339.18 -415.8  -1094.50    96.54   range=chr3R:2175000-2175999 (-40.06 -415.79 -1094.95 96.39 agagctccgggaagcactacggcgtgtcctgctgcgatgggtgctcctgc); range=chrX:13500000-13500999 (-11.51 -135.40 -361.01 72.93
140 range=chr3R:19000-19999 952 50  -324.14 -398.40 -1060.80    92.01   range=chr3R:2176000-2176999 (-34.21 -398.39 -1061.29 91.85 tatcctgggaagccactctaaagccgccctgatctccttagcccgctaca)  TATCCTGGGAAGCCACTCTAAAGCCGCCCTGATCTCCTTAGCCCGCTACA
141 range=chr3R:19000-19999 460 50  -325.80 -401.20 -1077.10    89.16   range=chr3R:2175000-2175999 (-31.29 -401.19 -1077.96 88.88 tttatggagtccctaatcctgtgcagaaaaggtaaagctccgcctgtgcg)  TTTATGGAGTCCCTAATCCTGTGCAGAAAAGGTAAAGCTCCGCCTGTGCG
142 range=chr3R:19000-19999 343 50  -342.85 -420.59 -1110.70    95.50   range=chr3R:2175000-2175999 (-39.29 -420.60 -1111.20 95.34 cgagcgtcccactggtctctggacatcagcgccatgatcgacggctgcgg)  CGAGCGTCCCACTGGTCTCTGGACATCAGCGCCATGATCGACGGCTGCGG
143 range=chr3R:19000-19999 272 50  -332.82 -409.29 -1092.50    91.41   range=chr3R:2175000-2175999 (-34.08 -409.29 -1093.43 91.11 agttcgctctgttggatcgctgccaacaagacgccatctttcaggtggtg)  AGTTCGCTCTGTTGGATCGCTGCCAACAAGACGCCATCTTTCAGGTGGTG
144 range=chr3R:19000-19999 207 50  -337.82 -414.5  -1095.30    95.12   range=chr3R:2175000-2175999 (-38.45 -414.49 -1095.85 94.94 gctgcacttccagatcctcgcccagatccttgtcacgtgcctgcgccagg)  GCTGCACTTCCAGATCCTCGCCCAGATCCTTGTCACGTGCCTGCGCCAGG
145 range=chr3R:19000-19999 7   50  -339.70 -416.29 -1094.2 97.08   range=chr3R:2175000-2175999 (-40.65 -416.29 -1094.69 96.92 gggaactgtgtggtggacaaggcgcggcggaactggtgtccctcctgccg)  GGGAACTGTGTGGTGGACAAGGCGCGGCGGAACTGGTGTCCCTCCTGCCG
146 range=chr3R:20000-20999 952 50  -327.24 -402.7  -1077.90    90.26   range=chr3R:2177000-2177999 (-32.49 -402.69 -1078.83 89.95 agccaaggcatattgtacttgcagtgagttcaagtggagagtggtggccg)  AGCCAAGGCATATTGTACTTGCAGTGAGTTCAAGTGGAGAGTGGTGGCCG
147 range=chr3R:20000-20999 860 50  -334.75 -411.99 -1103.50    90.25   range=chr3R:2176000-2176999 (-29.21 -337.99 -899.85 90.26 tggagaagtggtgtccacgcgaatccgtgaatcgcaagcgt)    TGGAGAAGTGGTGTCCACGCGAATCCGTGAATCGCAAGCGTTTCTTCTTC
148 range=chr3R:20000-20999 808 50  -336.14 -412.79 -1095.10    93.68   range=chr3R:2176000-2176999 (-36.72 -412.79 -1095.95 93.40 tgtgtgcgagtgcggccaagtgttgtgggctcggaaaactccaagaagcc)  TGTGTGCGAGTGCGGCCAAGTGTTGTGGGCTCGGAAAACTCCAAGAAGCC
```

In this oligo.txt file, oligos with multiple putative targets are reported along with all the putative target sites; an example of this can be seen in line 138 above. We have a written a script, clean_file.py, that processes oligo.txt files and removes probes with multiple putative targets. clean_file.py outputs a "cleaned" oligo.txt file, which contains detailed information about each oligo, as well as a .bed file in the same format as the Oligopaints .bed files hosted on the Oligopaints website. A screenshot is shown below:



```
7495  chr3R  997287  997318  TGACTGCTATACTAAATTTGCCACACTTAGCCTAGGCCTAGACTGAAACT  85.37
7496  chr3R  997160  997191  ACACGGCTAGCGATCCTGATCAAGATTATATGGTAAAAGGAAGCGTTACT  85.49
7497  chr3R  998775  998806  AGTGAACTTGGCCCAAACGCATTACCGTTAGAAGCGCCTAATTGCCCAAA  89.90
7498  chr3R  998723  998754  CAAATGGGCCAGACAGATAAACACACAGTTCCAGCTACCAATGAGCCGGC  89.79
7499  chr3R  998585  998616  CCTCTCCCAACTTTTCCCATTATGATTTCCTGGCCAGCAAGCTATTATAA  85.41
7500  chr3R  998434  998465  AGGGTTATCGTTAAGGGCGGCACAATTGTACATAATTGATGATACACCAA  85.28
7501  chr3R  998336  998367  TCGCACTCCGCAAATTTTGACATTAATTACTCATCAAATTGTGGCCACAT  85.33
7502  chr3R  998050  998081  GTGAGGAAATTAGTTTTGGCGGTTTGCGAACCTAACGCACAGGGTGTTAG  88.01
7503  chr3R  999901  999932  AAGCGAGCAAATGGCAAACGCTGCATGTAAGCACTAAATCAAAGTGTCAA  87.44
7504  chr3R  999840  999871  TCGAACGGCAACAAAGTTGGGCTCCAAATAGATATATCAGCAGCTTTAAA  85.48
7505  chr3R  999788  999819  GGCTGACATAAGTTGACAAAGTTTTCCAGCCGCGGCAGTCTGATTCGCTC  89.83
7506  chr3R  999578  999609  TGGCATTGCTGCATATTATGCGAGTAATAGCAAGGAAGAAGTTCGTTAGA  85.11
7507  chr3R  999519  999550  GGCACACGGCTATGTCATCTTTCACTTAACTTCACTCTCAACTGATGAAA  85.27
7508  chr3R  999390  999421  TGTTGCTGCTGTTAGATACACACGCAGGCAATAATCATAGGCCACTCATT  87.31
7509  chr3R  999246  999277  ATTACAGCCATAGCGCATAGAACTTGGCCCGCTCGTCTATCATTAGCATC  87.90
7510  chr3R  999142  999173  ATTGATTGCACGATTATCGCCAAATACAGACAAAGACAGACGGCGGTTTT  86.82
7511  chr3R  999084  999115  GTGCTGCTGTGCCTCTGCTGCATATCCGAACGATTGCATTAAATAGCAAT  88.03
7512  chr3R  999029  999060  CGAGTTCGGGAAACTAACTGTTCATCTTTGATTTTGCATAATCGCTGCGA  85.78
```

At this point, this file is now compatible with all of the scripts written for working with Oligopaints .bed files. Our probe mining has discovered 7512 probes over this 1 Mb region, giving a probe density of 7.5 probes/kb. At this point, we could sort these probes by T_M using the sortFile.py script, or use probeNumber.py to find the densest 2500, etc. We will make the /hox_example/ directory and all of its contents available as a .tar.gz on the Oligopaints website so users can try to replicate this example on their home systems/clusters. Good luck and happy FISHing!