created by V. Rich, graduate student in the DeLong Lab
last modified 2/12/08

Generating oligonucleotide probes for the marine microbial microarray:

all instructions are for a Mac set-up

Step 1: Generate a fasta file with all genes from each genome/genome fragment:

There are a variety of ways to do this. I now use a perl script. However, an easy way to do one or a few sequences is via the program Artemis.

- launch **Artemis**

within a Terminal window, cd to the Directory containing Artemis and then type "art" and return – if this doesn't work, type "csh", return, then try again. Artemis should launch. If you search for Artemis in your computer, you may find that you have an icon-launchable version of it.

1. Options -> click off eukaryotic mode
Open -> File -> file of choice, gb file (yes, ignore the error)

2. Select -> CDS features
Write -> bases of selection -> fasta format
use a ".fna" file extension, with appropriate prefix

Naming conventions: Location with some indication of clone type & depth if possible, and coordinates, eg. HF10_04G06 is Hawaii fosmid 10m, library plate 4 coordinates G06. For files, the convention is that ".fna" is used for fasta nucleic acids, and ".faa" is used for fasta amino acids

3. Select -> All
Write -> bases of selection -> fasta format

4. Click in the lower gene list window and hit
<control> and mouse click, then -> Save List as

This list is useful for annotating your oligos and for quickly checking the gene content of a fosmid/BAC later without having to launch Artemis. Some versions of Artemis won't let you do this; if not, no worries.

- then open the ".fna" file in **BBEdit** (or a similar text-editing program, one which won't add a bunch of stuff like Word does).

clean up file names as needed so that they lack spaces or funny characters.

This will be a different process from file-to-file since for many in-house sequences we're still working with unpublished versions that may not be perfectly named, etc. Generally, I prefer to keep each gene name as the CloneIdentifier_GeneIdentifier – often times this will be the sequence location of the gene because the gene names haven't yet been added, or it may just be CDS_001, _002, etc., depending on what information the Artemis-parsed CDS list contains. An ideal naming would be, eg., AntFos04D03_0to633, meaning AntFos library clone 04D03, the CDS from 0 t0 633.

- copy the file into the ArrayOligoSelector folder

Step 2: Use Array Oligo Selector to generate the potential probes:

ArrayOligoSelector is available, along with all documentation, at
http://arrayoligosel.sourceforge.net/
If working on a Mac you will need to download the version of formatdb and blastall from the NCBI website whose date corresponds to the same release date (or as close as possible) of the AOS version you're using, because the bundles AOS download comes with a Linux-compatible version of formatdb and blastall. If you're doing more complicated things with AOS than what is described below, there will also be other things you would need to download separately to allow complete functionality of the program, but for the scripts we run, this is sufficient. I have compared the results of AOS set up this way on a Mac Powerbook G4 to those from AOS set up on a Linux machine and they are identical.

- in the Terminal window, change into the ArrayOligoSelector directory

```
% cd [drag and drop ArrayOligoSelector folder for pathname]
```

- script 1 generates a list of all possible oligos from the input sequences, in a sliding window manner. The output file is "output 0", you can view it as text.

```
% pick70_script1
```

if this doesn't work, try typing ./pick70_script1
if you just type this you'll get a USAGE error telling you exactly what parameters you need to input:

*inputseq: gene/NUCLEOTIDE sequences submit for design in FASTA format

*genome: genome GENE/NUCLEOTIDE sequences in FASTA format

*oligo_size: in basepair

*MaskByLowercase: You can exclude sub-sequences from the compuation using lower case. Those sub-regions will be flagged in the outputs. To use lowercase for this purpose, type "yes"; otherwise, type "no".

In the case of this array, we're choosing 70-mers, and are not doing any masking of sequence.

So, what we'd really like to do is:

```
% pick70_script1 <input>.fna <input>.fna 70 no
```

For historical reasons, we use the same CDS output fna file as both CDS file and genome file, against which ArrayOligoSelector checks for uniqueness.
We discussed a dizzying array of possibilities for what to use as the genome file: concatenating all the fosmid and BAC sequences, using all the prokaryotic sequences in the nr database – these days one could imagine using all available environmental sequences as a "genome"… BUT, for our purposes, we did the simplest variant possible – using the CDs file as CDS and genome. For different organisms there is different coverage of the nearby related "sequence space", and this coverage changes all the time. One could try to make an array with much more specific probes, even to the point of doing alignments, etc, as other groups have done for other arrays, but that's not the purpose of this array. The goal was to see whether a "blind" design approach would allow discrimination among related genotypes, and with the prototype array I demonstrated that it did. If one were designing a different array for different purposes, or a different system, one might want to use a different design strategy.

Script 1 will show a warning error because it can tell we're not running Linux and they want to make sure we've got the correct versions of blastall and such

installed, and python – even if everything is good, it still gives you this warning - so type "yes" to proceed when queried.

- script 2 chooses among the many possible oligos for each gene to give you the ones closest to your desired parameters.

```
% pick70_script2
```

again, if this doesn't work, try typing ./pick70_script2
again, if you just type this you'll get a USAGE error telling you exactly what parameters you need to input:

*GC: GC percentage (eg: 35.5, positive float or integer number)

*Oligo_len: length of Oligo in bp(positive integer)

*Number_Oligo: how many oligos do you want to design (positive integer)

*OPTIONAL binding energy cutoff: 0 is the default

*OPTIONAL masking parameters: if used, all the optional masking parameters are required

*Mask_Length: maximum length of subsequence allowed containing the Mask_Symbols eg: 20

*Mask_Symbol (ATGCN): masking bases eg:AT or N

*Mask_Tolerance (0 -1) : percent of other bases allowed eg:0.1

So, we'd like to do 40% GC content (which was the average GC content of the few tens of fully-sequenced clones present in the lab database at the time I started this), and 70-mers, and 1 probe per gene, with no binding energy cutoff and no masking:

```
% pick70_script2 40 70 1
```

- copy the oligodup and oligofasta output files from the ArrayOligoSelector folder into a new location (remember, ArrayOligoSelector has to rewrite those intermediate files each time, so you have to save them before you can run it again), and rename the files based on the clone/organism name.

Step 3: Choose which output oligos to use as probes:
Again, there are different ways to process the AOS outfiles… I now use a perlscript to do this, which will get posted on the website too, but this is a simple, alternate way to do the same thing manually.

- open either output files in BBEdit, select all, and <apple> <F> to find and replace – click on the lower left box for a Multifile search to include the other file, and use grep:

find: \r
replace: (just a blank space)

then

find: >
replace: \r>
save files

- open both files in Excel with a space as column delimiter.
merge the files into one

sort by %GC (column D or E, depending)

if there are <20 oligos with 40%GC, then take those just higher and lower until you have 20. Highlight these 20 oligos – these are your probes.

if there are >20 oligos with 40%GC, then sort *among those* oligos by ΔG of hybridization (column G usually), and take the 20 oligos with the lowest (=most negative) ΔG values, within those that have 40%GC. Highlight these 20, these are your probes.

ΔG has been shown to correlate inversely with hybridization signal for microarray probes, which makes good sense – so if you've got a surfeit of potential probes with the "right" %GC, ΔG makes a good criterion for selecting among them!

Copy and paste your chosen oligos into your master oligo file, and proceed as you see fit.

An **important thing to note** here is that "blind" probe design means that the process outlined above does *not* targeting particular genes of interest.