File S1

Scripts written for BLAST

We BLASTed against the entire non-redundant protein database, which we downloaded from NCBI as well (ftp://ftp.ncbi.nih.gov/blast/db/ - five files on the website, “nr.00.tar.gz” through “nr.04.tar.gz”). Here is the call we used:

```
blast-2.2.18/bin/blastall -p blastp -i input.fa -d data/gb/nr -C 2 -m 8 -o output.blast
```

Our input files were either the entire *C. elegans* genome, from two sources (NCBI: http://www.ncbi.nlm.nih.gov/protein search ‘ "Caenorhabditis elegans"[porgn:__txid6239] ’ or Wormbase (Wormpep 220: http://www.sanger.ac.uk/Projects/C_elegans/WORMBASE/current/wormpep.shtml), or a list of 29 newly-identified genes found in this paper.

We analyzed our output using the following script (the actual script is in grey text, the black text are annotations). The script analyzes BLAST output, giving the user the number of times a particular gene ‘hit’ either a nematode or *C. elegans* gene, by using a hash created from reference files of all nematodes (NCBI: http://www.ncbi.nlm.nih.gov/protein search ' Nematoda [ORGN] ') or *C. elegans* (see above).

Analysis Script:

```
#!/usr/bin/perl
```

Step 1: We created a hash of the gi numbers of *C. elegans* genes ($Celegans), of nematode genes ($nematodes), and a combined hash of both ($total). We created our hash by reading in the NCBI *C. elegans* genome and extracting the gi numbers from each sequence. We repeated this for the nematodes.

```
$Celegans=();
$total=();
$filename= "NCBI_Cel2.17F2.fa";
open (DATA, $filename) || die "where is $filename ?\n";
until (eof DATA) {
        $line = <DATA>;
        chomp $line;
        @array = split('\|', $line);
        $Celegans{$array[1]} = $array[1];
        }

$nematodes=();
$filename= "Nems10.26F2.fa";
open (DATA, $filename) || die "where is $filename ?\n";
until (eof DATA) {
         $line = <DATA>;
         $line1 = <DATA>;
         $line2 = <DATA>;
         chomp $line;
         $line =~  />gi\|(\d+)\|/;
         $nematodes{$1} = $1;
        }

%total = (%Celegans, %nematodes);
```

Step 2: Our program read in a line of data, analyzed it, and went on to the next. Because of this, we needed to add an arbitrary line to the end of the file, so that the last line would be analyzed. Here we added a string of X’s to the file.

```
$filename = "input.blast";
open ADDEND, ">$filename.end";

open (PERL, $filename) || die "where is $filename ?\n";
until (eof PERL) {
        $line = <PERL>;
        chomp $line;
        print ADDEND $line."\n";
        }

print ADDEND "XXXXXXXXXXXXXXXXXX \n";

close(ADDEND);
close(PERL);
```

We wanted data on hits for each individual gene, so we created a temporary file for all of its data to be added to. We also needed to create an output file for our hash analysis.

```
unlink "temp.txt";
open FILE, ">>temp.txt";

$filename2 = "$filename.end";
open OUTPUT, ">>HITS.$filename";
open(BLAST, $filename2) || die "where is $filename2 ?\n";
```

Step 3: We began our main analysis loop. We read in the first line of BLAST data, extracted the gene name, printed its data to our temporary file, and set our counters—nematodes to 0, and *C. elegans* to -1 (to correct for self-hits).

```
$line = <BLAST>;
chomp $line;
print FILE $line."\n";
@firstline = split('\t', $line);
$firstgene = $firstline[0];
$nemas = 0;
$cels = -1;
```

Step 3a: We extracted the gene name for the next line of data, compared it to the previously stored gene name, and if it was the same, we printed the full line of data to our temporary file. We repeated until we encountered a new gene.

```
        until (eof BLAST) {
                $line = <BLAST>;
                chomp $line;
                @nextline = split('\t', $line);
                $gene = $nextline[0];

                        if ($gene eq $firstgene) {
                                print FILE $line."\n";
                                } else {
```

Step 3b: When we have encountered a different gene name, it means we have written all of the BLAST data from a particular gene into our temporary file. We close and reopen that file, to begin analysis on it from the beginning. First, we pulled out the gi number of the nr database sequence that it hit, and designated a new hash named $match.

```
                                close(FILE);
                                open(FILE, "temp.txt") || die "where is temp.txt?\n";
                                $match = ();
                                $seq = <FILE>;
                                chomp $seq;
```

```
@tab = split('\t', $seq);
@array = split('\|', $tab[1]);
$gi = $array[1];
```

Step 3c: Until the end of our temporary file, we took a line's gi number, and compared it to $match. If it existed in $match, we skipped it. This ensured we didn't count multiple hits to the same gene as hits to different genes. If our gi wasn't in $match, we compared it to $total. If it was in $total, we added a value to our nematode counter. We then compared it to $celegans—adding a value to our *C. elegans* counter if it was present. After this, we added that gi number to $match.

```
until (eof FILE) {
        if (exists ($match{$gi})) {
        }else {
        if (exists ($total{$gi})){
                $nemas++;
                }
        if (exists ($Celegans{$gi})){
                $cels++;
                }
        }
$match{$gi} = $gi;
$seq = <FILE>;
chomp $seq;
@tab = split('\t', $seq);
@array = split('\|', $tab[1]);
$gi = $array[1];
}
$sum = ($nemas+$none+1);
if ($cels == -1) {
        $cels = 0;
        }
```

Step 3d: We printed to the output file the name of our gene and the counter values. We then reset the counters, cleared the $match hash, and deleted the temporary file. The loop began again until the entire BLAST file was analyzed. All output of this loop was added to the same output file.

```
print OUTPUT $firstgene."\t"."Nematode hits: $nemas\t"."Cel hits: $cels\n";
        $nemas = 0;
        $cels = -1;
        %match = ();
        $firstgene = $gene;
        unlink "temp.txt";
        open FILE, ">>temp.txt";
        print FILE "$line\n";
        }

    }

exit;
```